ST COMPUTER

Die Fachzeitschrift für den ATARI-ST Anwender.

Januar 86
DM 6,–
NR. 1

* Komplette Basic-Syntax zum Heraustrennen
* 1 Mega Byte im Selbstbau
* 5¼ Zoll Floppy am ATARI ST
* Neue Software, Berichte, Tests
* Hardware des ST

Premierenausgabe
260 ST, 520 ST, 520 ST+



* Logo-Kurs * Grafik-Programm zum Abtippen



ST – eine neue Fachzeitschrift

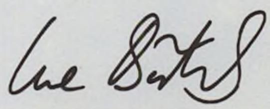
Liebe Leser!

Die Redaktion der ST freut sich, Ihnen die Erstausgabe einer Fachzeitschrift für Ihren Atari ST präsentieren zu können.

Die Ziele der Zeitschrift liegen darin, daß Sie einen Einblick in die Rechnerbedienung, in die Rechnertechnik und in das Betriebssystem des Atari ST bekommen. Ferner werden wir Sie über Neuheiten auf dem Soft- und Hardwaresektor informieren. Diese Produkte werden von unseren Mitarbeitern ausführlich getestet und Ihnen präsentiert. Außerdem werden in monatlichen Abständen Kurse in den wichtigsten Programmiersprachen erscheinen (Logo, C, Pascal, Fortran und Assembler), sowie spezielle Anwendungen in Basic.

Die Zeitschrift behandelt neben der Theorie vor allem die praktische Seite, um es Ihnen zu ermöglichen sämtliche Anwendungen nachzuvollziehen.

Die Redaktion und Ihre Mitarbeiter stehen selbstverständlich für Fragen und Auskünfte zur Verfügung.



Eine neue Rechnergeneration

Seit einigen Monaten ist ein neuer Personalcomputer auf dem Markt, der Atari ST. Es gibt ihn in zwei Ausführungen die sich lediglich in der Größe des internen Ram's unterscheiden. Der 260 ST mit 512 KByte Ram und der 520 ST+ mit 1 MByte sind also weitgehend softwarekompatibel. Für den 520 ST gilt dies genauso da er mit dem 260 ST, abgesehen von einem fehlenden Videosignal, völlig identisch ist.

Es kann der ST Serie schon heute eine große Karriere vorausgesagt werden. Die Gründe dafür liegen hauptsächlich an den technischen Merkmalen und natürlich auch am Preis, denn in dieser Preisklasse ist nichts Vergleichbares vorhanden.

Der Rechner arbeitet mit einer 68 000 CPU von Motorola, dem momentan leistungsfähigsten Microprozessor im Personalcomputerbereich. Er hebt sich durch seinen großen Befehlssatz (z. B. implementierte Multiplikations- und Divisionsbefehle), durch seine 16/32 Bit-Struktur und die damit verbundene hohe Rechengeschwindigkeit von den bekannten Microprozessoren ab.

Mit einer Speicherkapazität von 512 KByte liegt selbst der kleinere ST-Rechner weit über vergleichbaren Personalcomputern (IBM, Macintosh, Amiga).

Durch das neuartige GEM (Graphics Enviroment Manager) wird dem Benutzer eine leichte und komfortable Zugriffsmöglichkeit auf die rechnerinternen Möglichkeiten geboten. So entfallen die manchmal sehr komplizierten Befehle die an eine exakte Syntax gebunden sind, und werden durch anschaulichen, leicht erlernbaren grafischen Bedienungssymbolen ersetzt.

Schon jetzt gibt es eine große Anzahl von hochwertiger Anwendersoftware, wie Textverarbeitungs, Datenverwaltungs- und Kalkulationsprogramme. Außerdem sind mittlerweile die gängigsten Programmiersprachen erhältlich. Dazu zählen verschiedene Basicversionen, Pascal, C, Logo, Fortran, Modula-2, LISP, BCPL und Assembler.

Selbst rechnerunterstützte grafische Konstruktion (CAD) ist, wenn auch nicht in Perfektion, so doch zumindest ansatzweise auf diesem Rechner möglich.

Alle diese Gründe sprechen für den Atari ST der sowohl für den professionellen Benutzer als auch für den Hobbyprogrammierer ein leistungsstarkes Gerät darstellt.

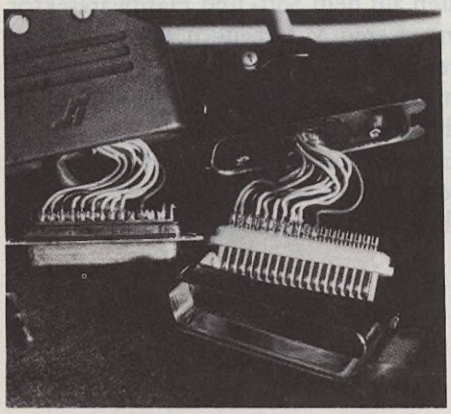
Inhalt

Allgemeines

Editorial 1

Merkmale und Unterschiede des 260 ST, 520 ST und des 520 ST+ 3

Hardware

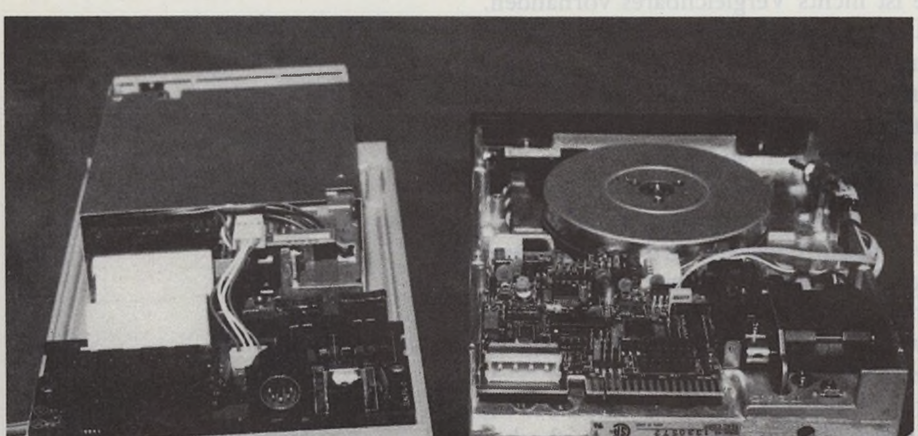


Ein Druckerkabel für 30,– DM 4

Funktioniert ein gewöhnlicher Monitor am ATARI ST? 5

Die Hardware des ATARI ST 7

Anschluß eines 5 1/4 Zoll Laufwerks an den ATARI ST 10



Grundlagen: RAM-Bausteine 13

Ein Mega-Byte im Selbstbau 15

Software

Software für den ATARI ST 18

ST-Textmanager 22

BASIC-Syntax 27

Das neue ST-BASIC – ein Basic der Superlative? 39

C-Text & C-Adress 43

Kuchen- & Säulengrafik Listing 45

Kontrolliertes Chaos Listing 61

Kurs

LOGO oder eine Einführung in Turtle-Geometrie 56

Aktuelles

Einkaufsführer 52

News & Infos 62

Bücher 63

Vorschau 64

Impressum

ST-Computer

Herausgeber: Heim Fachverlag, Heidelberger Landstraße 194, 6100 Darmstadt 13, Telefon (0 61 51) 5 53 75

Redaktion: Uwe Bärtels (Chefredakteur), Harald Schneider, Marcelo Merino, Harald Egel

Freier Mitarbeiter: Markus Nerding

Titelseite: Baumann, Blume & Partner

Anzeigen: Anzeigenleiter H. Heim, Telefon (0 61 51) 5 53 75
Anzeigenpreise nach Preisliste Nr. 1 gültig ab 1.1.86

Erscheinungsweise: 11 x jährlich

Bezugspreis: Einzelheft DM 6,–. Jahresabonnement DM 60,– inklusive der gesetzlichen Mehrwertsteuer und den Zustellgebühren für 11 Ausgaben.

Bezugsmöglichkeiten: ATARI-Fachhändler, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag unter obiger Adresse.

Druck: Ferling Druck Darmstadt

Manuskripteinsendungen: Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern im Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

Merkmale und Unterschiede des 260 ST, 520 ST und des 520 ST+

Es ist schon auffallend, daß ATARI kaum ein halbes Jahr nach der Einführung des 520 ST zwei weitere Modelle dieses viel gefragten Rechners auf den Markt brachte. Bemerkenswert dabei ist, daß der ohnehin schon so preiswerte 520 ST vom 520 ST+ abgelöst wurde, trotz Verdoppelung des RAM Speichers der Preis aber identisch blieb. Diese an für sich positive Entwicklung des Mega ATARI's ist natürlich für diejenigen eine schmerzliche Erfahrung, die den „alten" 520 ST kurz vor der Bekanntmachung auf der Systems 85 in München (Ende Oktober) gekauft haben. ATARI begründet diesen schnellen Fortschritt, einerseits mit dem gesunkenen Dollarkurs und den damit verbundenen günstigeren Rohstoffpreisen, andererseits ist die Entwicklung in der Computer-Branche schon immer sehr zügig vorangegangen und somit kaum vorhersehbar.

Der wesentliche Unterschied der drei neuen ATARI-Rechner, liegt wie schon erwähnt in der Speicherkapazität. Wobei der „alte" 520 ST und der darauf in München neu vorgestellte 260 ST jeweils über 512 KByte RAM-Speicher verfügen. Dieser umfangreiche Speicher ist nun durch den 520 ST+ auf 1 024 KByte! (= 1 Mega-Byte) erweitert worden. Der Speicher schrumpft allerdings relativ schnell, da das TOS mit seiner grafischen Benutzeroberfläche GEM schon über 200 KByte verschlingt. Außerdem sind viele der heutigen Programme nicht mehr in Assembler (Maschinencode) geschrieben, sondern meist in einer Hochsprache (wie z. B. „C"). Dies hat den Vorteil, daß fertige Programme schnell und leicht an neue Rechner angepaßt werden können, allerdings auch Nachteile wie größerer Speicheplatzbedarf und geringere Ablaufgeschwindigkeit. So bleibt beim ST-Basic (erste Version) nur gut 10 KByte freier Speicher übrig. Aus diesem Grund ist ein relativ großer Speicherplatz nicht verkehrt. Natürlich wird trotz allem ATARI's Speicher-Riese durch Programme kaum zu füllen sein, insbesondere wenn Anfang nächsten Jahres das TOS im internen ROM verschwindet. Es sei allerdings bemerkt, daß für aufwendige Grafiken, wie z. B. bei CAD-Anwendungen, auch ein Mega-Byte Speicher schnell verbraucht ist.

Im Prinzip sind die drei Rechner voll kompatibel sofern der Speicherplatz der „Kleineren" ausreicht und die Software nicht direkt auf das TOS bzw. den Bildschirmspeicher zugreift. Im Prinzip deswegen, weil der Rechner TOS und Bildschirmspeicher immer an das obere Speicherende lädt bzw. schiebt. Greift nun die Software des „alten" 520 ST (260 ST) z. B. direkt (absolut) auf den Bildschirmspeicher des 520 ST+ zu, so findet das Programm dort keinen Bildschirmspeicher mehr, sondern nur leeres RAM. Deshalb laufen solche Programme nicht. Dieser Fehler ist jedoch auf schlechte Programmierung zurückzuführen und tritt recht selten auf.
Zwischen dem „alten" 520 ST und dem neueren 260 ST gibt es bis auf eine Ausnahme keinen Unterschied. Diese Ausnahme betrifft die Monitorausgangsbuchse. Beim 520 ST ist der Pin zwei des Monitoranschlusses nicht belegt, obwohl in manchen Handbüchern dieser Pin mit „Composite Video" bezeichnet wurde. Beim neueren 260 ST ist dieser Anschluß mit „Composite Synchron" belegt. Dies ist ein Mischsignal aus dem horizontal- und vertikal Synchronimpuls. Zusätzlich ist Pin acht nicht mehr auf Masse (520 ST), sondern über einen 1,2 KOhm Widerstand auf plus 12 Volt gelegt. Mit diesen Änderungen (siehe entsprechende Anleitung in diesem Heft) kann nun direkt ein modernes Farbfernsehgerät mit SCART-Eingang angeschlossen und als Monitor verwendet werden. Dabei ist jedoch nur niedrige und mittlere Auflösung möglich. Wer sich diese kleine Änderung in seinen „alten" 520 ST einbaut, ist dann stolzer Besitzer eines neuen 260 ST. Läßt man sich nun noch den RAM-Speicher auf ein Mega-Byte erhöhen, erhält man einen 520 ST+. Bastler können dies übrigens auch selbst tun. Wie – das steht an einer anderen Stelle in diesem Heft.
Entgegen der Ankündigungen in einigen Fachzeitschriften, ist bei allen drei modellen kein HF-Modulator eingebaut. Ein solcher HF-Modulator würde die Möglichkeit bieten über den Antenneneingang einen Fernseher anzusteuern und somit ein Computerbild auf ihm darzustellen. Dies wäre jedoch wegen der begrenzten Videobandbreite des Fernsehers (unter 6 MHz) nicht sinnvoll. Denn die Videobandbreite des Rechners beträgt bei hochauflösender Bildschirmdarstellung über 32 MHz! Bei niedrigerer Auflösung immerhin noch etwa 7,5 MHz. Wer also ein Bild wünscht, bei dem man jede Feinheit klar erkennen kann und der über die nötige Finanzkraft verfügt, der sollte auf einen (Farb-) Monitor nicht verzichten.
Sie sehen, alle drei Rechner sind technisch gesehen kaum verschieden – zum Glück, wie wir meinen. Allerdings kann man sie in unterschiedlichen Konfigurationen erwerben. So dürfte der „alte" 520 ST nicht mehr zu kaufen sein, an seine Stelle ist der 520 ST+ getreten. Dieser wird zur Zeit nur komplett, einschließlich Monochromen-Monitor (SM 124) und 3 1/2 Zoll Floppy (SF 354 mit 360 KByte Speicherkapazität formatiert) sowie der Maus zu einem wirklich ungewöhnlich niedrigen Preis (ca. DM 3 000,–) verkauft. ATARI will mit diesem Modell vor allem den professionellen Anwender ansprechen. Als „Aufsteiger" Modell bezeichnet ATARI den 260 ST. Dieser Rechner ist einzeln erhältlich und kostet ca. DM 1 300,–. Damit ist der 260 ST für all diejenigen interessant, die bereits einen RGB-Farbmonitor oder einen Farbfernseher mit SCART-Buchse besitzen. Man benötigt dann allerdings noch eine Maus und eine Floppy, wobei man nicht unbedingt an die ATARI Floppy gebunden ist. Man kann somit verhältnismäßig preiswert in die neue 16-Bit Rechnergeneration einsteigen. Möchte man aber mit der hochauflösenden Grafik arbeiten, muß man den ATARI schwarz-weiß Monitor (und nur dieser ist wegen der höheren Zeilenfrequenz und des daraus resultierenden scharfen Bildes anschließbar) benutzen. Damit liegt man jedoch nur noch ca. DM 300,– unter dem Preis des großen Bruders.

Ein Drucker kabel für 30,– DM

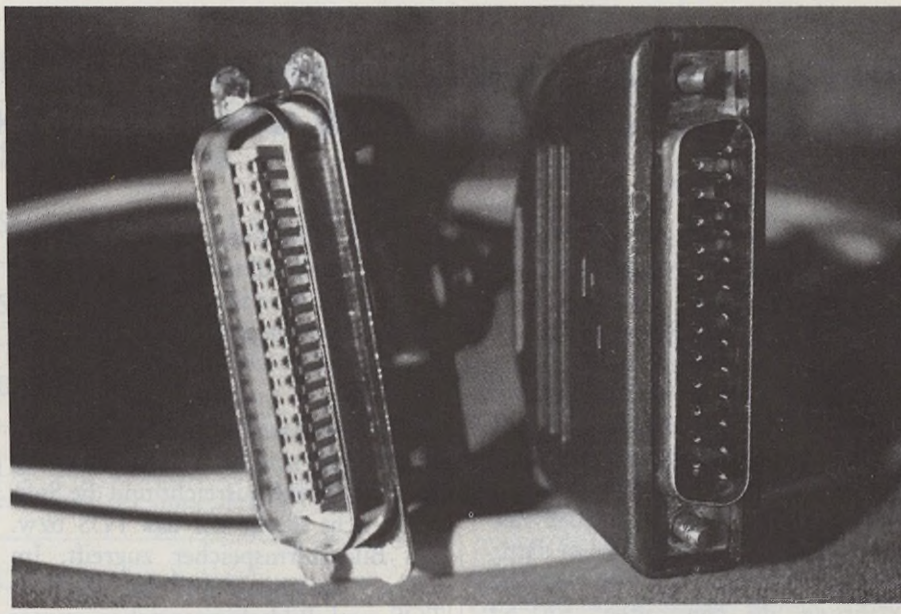


Bild 1: Das fertige Druckerkabel

Was nützt einem der teuerste Drucker und der beste Rechner, wenn zur Verbindung kein passendes Druckerkabel vorhanden ist? Dieses doch so wichtige Bindeglied wird nur allzugern beim Kauf übersehen. Wer also seine Dokumente gern schwarz auf weiß haben möchte, muß sich dieses Kabel entweder teuer kaufen oder diesen Artikel lesen und selbst zum Lötkolben greifen. Man nehme dazu ein 1 bis 3 Meter langes, 10-adriges, abgeschirmtes Rundkabel (im Notfall tut es auch normales Breitbandkabel), des weiteren einen 25-poligen Cannon- und einen 36-poligen Amphenolstecker. Diese nach der folgenden Anleitung zusammengebaut und -gelötet, ergeben ein funktionstüchtiges Druckerkabel für unter 30,– DM. Die Bauteile sollten in jedem besseren Elektronikgeschäft erhältlich sein, wobei der Preis natürlich von der Kabellänge und nicht zuletzt von der Gewinnspanne des betreffenden Händlers abhängt.

Die untenstehende Tabelle zeigt wie die zehn Adern des Kabels an die beiden Stecker angelötet werden. Hierbei ist jedem Anschlußstift eine Zahl zugeteilt. Diese Zahlen finden sich auch direkt bei jedem Stift auf den beiden Steckern wieder.

Für die Masse ist keine Ader im Kabel vorhanden, sondern man benutzt zur Übertragung die Abschirmung des Kabels. Es ist auch nicht notwendig alle Massestifte anzuschließen. Nach dem Löten sollten die Verbindungen wie auf den Bildern 2 (Amphenol) und 3 (Cannon) aussehen. Diese Anschlußbelegung traf auf alle uns zur Verfügung stehenden Centronics-Drucker zu. Diese sollten allerdings über einen sogenannten IBM-Modus verfügen, da sich ansonsten ohne größeren Hard- bzw. Softwareaufwand keine Hardcopy des Bildschirms ausdrucken läßt (Dies ist ja beim ATARI ST jederzeit durch gleichzeitiges Drücken der Tasten Alternate und Help zu erreichen). Die Ausgabe von ASCII-Zeichen ist aber auch ohne den IBM-Modus möglich, so daß man sich nicht gleich einen neuen Drucker kaufen muß, falls keine Grafiken auf dem Papier erscheinen. Die meisten Geräte sind allerdings auch ohne besonderen Hinweise auf den Modus grafikfähig. Also einfach einmal ausprobieren!

Zum Schluß wollen wir noch kurz die einzelnen Signale erklären, die für die Kommunikation zwischen Rechner und Drucker sorgen. Da wäre zuerst das STROBE-Signal. Es dient zur Synchronisation der Dateneingabe in den Drucker und liegt normalerweise auf High-Pegel. Sollen nun Daten gedruckt werden, geht das Signal auf Low-Pegel und es werden die Daten in den Druckerpuffer eingelesen. Als nächstes folgen die acht Datensignale (DATA 0–7) und zuletzt das BUSY-Signal. Es zeigt den Zustand des Druckers an und liegt auf High-Pegel, wenn der Drucker arbeitet und keine Daten empfangen kann. Dies ist der Fall, wenn der Druckerpuffer voll ist und das Gerät arbeitet. Außerdem auch noch, wenn der Drucker im Zustand OFF LINE oder ein Fehler aufgetreten ist.

Anschlußbelegung

DRUCKER Amphenolstecker Stift	Signal		RECHNER Cannonstecker Stift	Signal
1	STROBE	------------	1	STROBE
2	DATA 0	------------	2	DATA 0
3	DATA 1	------------	3	DATA 1
4	DATA 2	------------	4	DATA 2
5	DATA 3	------------	5	DATA 3
6	DATA 4	------------	6	DATA 4
7	DATA 5	------------	7	DATA 5
8	DATA 6	------------	8	DATA 6
9	DATA 7	------------	9	DATA 7
11	BUSY	------------	11	BUSY
19-30	MASSE	------------	18-25	MASSE

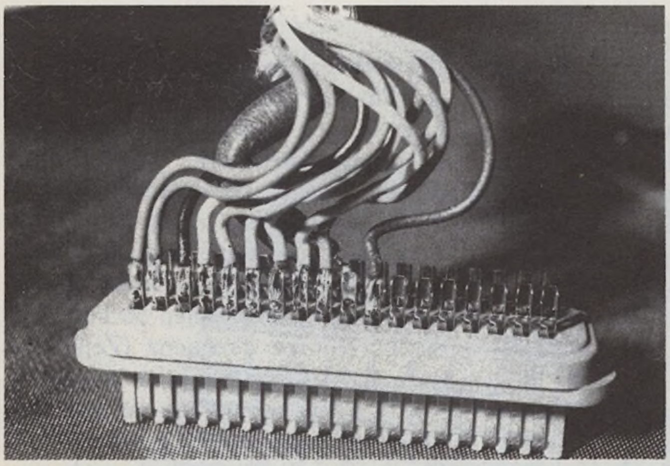


Bild 2: Amphenol-Stecker

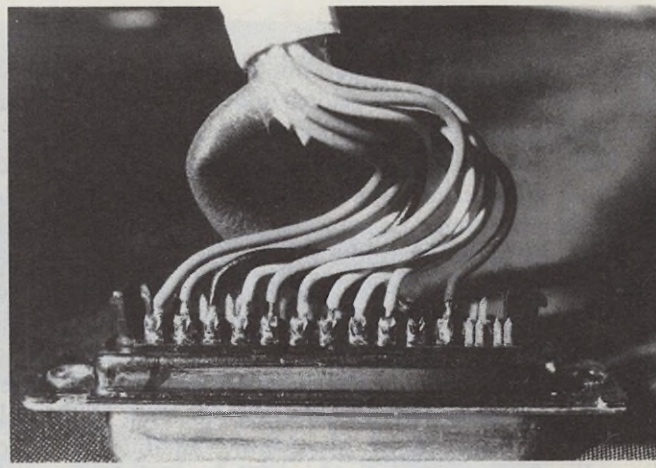


Bild 3: Cannon-Stecker

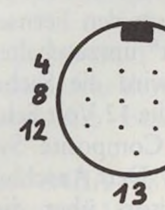
Da alle anderen Druckersignale nicht an den ATARI angeschlossen werden, gehen wir auf diese in diesem Bericht nicht weiter ein und wünschen „Gut Druck!".

Funktioniert ein gewöhnlicher Monitor am ATARI ST?

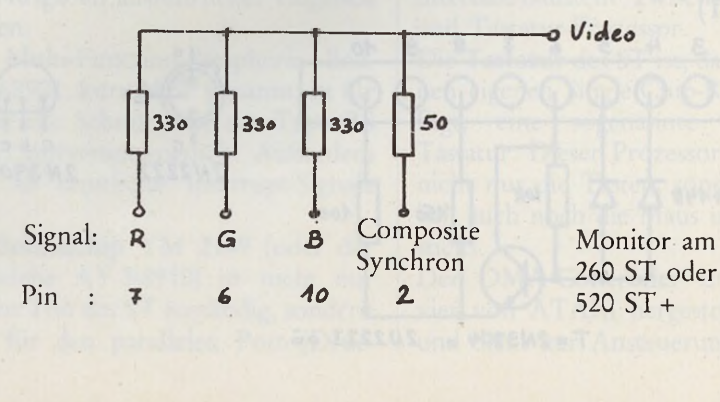
Um es gleich vorweg zu sagen, sie können mit einem Monitorstecker und vier Widerständen einen normalen Monitor oder auch einen Fernseher mit Video-Eingang anschließen. Bei unserem Test funktionierte ein kleiner schwarz-weiß Fernseher mit nachgerüstetem Videoeingang genauso gut wie ein moderner Grün-Monitor, sogar ein großer Farbfernseher mit Videorecorder-Anschluß erzeugte ein gutes Bild. Da die drei RGB-Ausgänge des ATARI ST nur über drei Widerstände á 330 Ohm gemischt werden, um die erforderliche Bild-Information zu bekommen (schwarz-weiß Bild), ist der hochauflösende Bildschirm nicht darstellbar. Zusätzlich zu den drei Farben (Rot, Grün und Blau) wird noch das Composite-Synchron Signal benötigt, dieses wird dann noch über einen 47 Ohm Widerstand zu den drei gemischten Farbsignalen hinzugemischt. Das so entstandene Video-Signal kann nun. über eine abgeschirmte Leitung (75 Ohm Kabel) einem normalen Monitor zugeführt werden.

Doch nun zur Praxis. Da der Monitorstecker des ATARI ST nur sehr schwer oder gar nicht zu bekommen ist, muß man sich einer anderen Technik bedienen. Zum Glück haben bei dieser Buchse die einzelnen Pins den genormten Abstand von 2,54 Millimeter zueinander. Dadurch hat man die Möglichkeit sich den Stecker selber zu bauen, indem man ein Stückchen Lochraster-Platine (ca. 20 x 20 Millimeter) mit den benötigten Pins bestückt. Als Anschlußpins kommen fünf dünne Lötnägel, die eine Länge von ca. 15 Millimeter haben müssen, in Frage. Diese werden dann am Besten in die benötigten Anschlüsse der Monitorbuchse gesteckt. Es sind die Pins zwei (Composite Synchron), sechs (Grün), sieben (Rot), zehn (Blau) und noch ein Pin für Masse (Pin 13). Nun wird die Lochraster-Platine auf die Enden der Lötnägel gesteckt und mit diesen von hinten verlötet. Leider fällt Pin 13 aus der Reihe, dieser Lötnagel muß ein wenig hingebogen werden. Beim Verlöten der Nägel mit der Platine ist auf kurze Lötzeiten zu achten, denn sonst könnte die Monitorbuchse des Rechners schmelzen. Somit hat man sich einen Monitorstecker gebaut.

Die Monitorbuchse



von hinten gesehen!



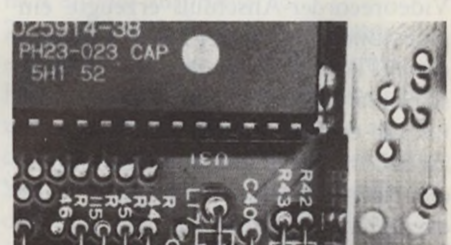
Jetzt werden die Widerstände mit einem Ende auf die Platine gelötet und zwar an Pin zwei ein 47 Ohm und an die Pins sechs, sieben und acht je ein 330 Ohm Widerstand (siehe Schaltbild unten). Die vier freien Enden aller Widerstände werden miteinander verlötet und mit dem Innenleiter des Monitorkabels verbunden. Der Außenleiter des Kabels wird mit Masse (Pin 13) verbunden. Das andere Ende des Kabels wird mit einem passenden Stecker für den Monitor versehen. Meist sind diese Eingänge mit „Video in (75 Ohm)" bezeichnet. Will man auch den Ton des Rechners hören, so muß auch noch der Pin eins (Audio-Ausgang) des Steckers bestückt werden. Dieses Signal kann direkt einem NF-Verstärker zugeführt werden.

Farbfernseher mit SCART-Buchse als Farbmonitor (auch am „alten" 520 ST)

Der oben beschriebene Vorgang, zum Anschluß eines normalen monochromen Monitors (hier ist nicht der ATARI Monitor SM 124 gemeint), ist nur beim ATARI 260 ST und 520 ST+ möglich, da diese neueren Modelle den Composite-Synchron Ausgang (Pin zwei) besitzen. Außerdem kann bei diesen Modellen ein modernes Farbfernsehgerät mit SCART-Eingang direkt als Farbmonitor verwendet werden, da auch plus 12 Volt am Pin acht vorhanden ist. Diese Spannung wird benötigt, um den Fernseher auf „Datensichtgerät" umzuschalten.
Beim 520 ST wird die Sache etwas schwieriger, da die 12 Volt Schaltspannung und der Composite Synchron Ausgang fehlen. Der Anschluß eines normalen Monitors über die RGB Ausgänge dürfte bei diesem Modell nicht so sehr interessieren, da ja ohnehin ein sehr guter schwarz-weiß Monitor beim Kauf dabei ist. Wer aber dennoch einen Monitor an seinen „alten" 520 ST anschließen möchte, der muß sich ein Composite-Synchron Signal (gemischter Synchronimpuls) erst erzeugen. Für denjenigen der einen kleinen Eingriff am Computer nicht scheut, besteht die einfache Möglichkeit dieses selbst, nach folgender Schaltung bzw. Foto, zu tun. Die kleine Schaltung mischt mittels der beiden Dioden das vorhandene horizontale- und vertikale Synchronisations-Signal. Der Transistor verstärkt das so entstandene Composite Synchron Signal und gibt es am Pin zwei der Monitorbuchse aus. Wegen der wenigen Bauteile ist eine freie Verdrahtung (wie beim Original auch) angebracht. Platz für die Schaltung ist genug vorhanden, da sie an die Stelle eingebaut wird, wo früher ein HF-Modulator geplant war. Dieser wird, wie schon erwähnt, aber nie eingebaut werden. Zum Anschluß eines Farbfernsehers mit SCART-Eingang als Farbmonitor fehlt nun nur noch die 12 Volt Schaltspannung. Diese wird am Besten direkt vom Netzteilstecker (Pin 4) über einen 1200 Ohm Widerstand zum SCART-Stekker geführt. Vorsicht, der Pin acht der Monitorbuchse des 520 ST liegt auf Masse und nicht auf plus 12 Volt, wie bei den neueren Modellen 260 ST und 520 ST+.

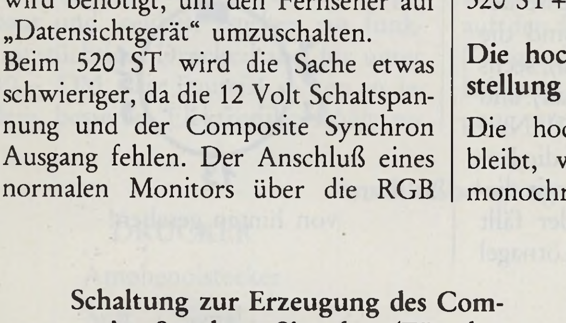
Die hochauflösende Bildschirmdarstellung

Die hochauflösende Bilddarstellung bleibt, vorerst zumindest, nur dem monochromen ATARI Monitor SM 124 vorbehalten. Erkennt der Rechner am Pin vier der Monitorbuchse ein Low (Masse), so schaltet er in die „monochrome Betriebsart". Man erkennt dies daran, daß z. B. die niedrige bzw. mittlere Auflösung nicht anwählbar sind. Da die vertikale Frequenz (Bildwiederholfrequenz) des Monitors 71 Hz (gegenüber normalen 50 Hz) und die horizontale Frequenz (Zeilenfrequenz) 35,7 KHz (gegenüber normalen 15,625 KHz) beträgt, ist das Bild des Monitors ausgesprochen scharf und flimmerfrei. Erklärt jedoch auch warum ein gewöhnlicher Monitor in der hohen Auflösung nicht funktionieren kann. Ein solcher Monitor hätte nämlich gerade die Hälfte einer Zeile „geschrieben", dann würde auch schon der Synchronimpuls für die nächste Zeile kommen.
Der Pin vier der Monitorbuchse (Monochrom-Sensor) ist durch einen internen Pull-up Widerstand auf High gelegt, so daß dieser Pin bei einem Farbmonitor unbenutzt bleibt. Erst durch den monochromen Monitor wird (wie oben erwähnt) dieser Pin vier auf Masse gezogen. Dies löst einen Interrupt im MFP-Baustein (68901) aus, was zu Folge hat, daß beim Initialisieren des Rechners die „monochrome Betriebsart" eingeschaltet wird. Hierin liegt auch der Grund, warum der Rechner beim Herausziehen des Steckers eines monochromen Monitors abstürzt.

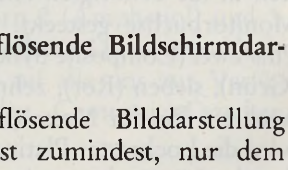


Aufbau der nebenstehenden Schaltung

Schaltung zur Erzeugung des Composite Synchron-Signales. (Für den 520 ST)



Anschlußbelegung: von unten gesehen





Die Hardware des ATARI ST

Mit dem folgenden Artikel wollen wir Ihnen einen Überblick über die gesamte Hardware verschaffen. Dabei wird jede wichtige Funktionsgruppe des Rechners erwähnt und mit ihren Aufgaben beschrieben. Es kann daher nicht die Aufgabe dieses Artikels sein, die Funktion jeder einzelnen Leitung des Schaltplans zu beschreiben. Den kompletten Schaltplan des Rechners kann man übrigens für ca. DM 30,– von ATARI beziehen.

Das nebenstehende Blockschaltbild zeigt einem in übersichtlicher Weise das Zusammenwirken der einzelnen Bausteine des ATARI ST Systems. Fast alle Komponenten sind an drei Busse angeschlossen. Unter einem Bus versteht man den Verbund von mehreren Leitungen, die zur Datenübertragung oder zu Steuerzwecken dienen. Die wichtigsten Busse sind im folgenden kurz erklärt. Da wäre zuerst der 16 Bit breite bidirektionale **Datenbus**, der für den Datentransport zwischen dem Prozessor und den übrigen Bausteinen zuständig ist. Des weiteren wäre da der **Adressbus**, der 23 Bit breit ist und nur in eine Richtung betrieben wird. Er wird vom Prozessor dazu benutzt, die Speicherchips, sowie auch andere Peripheriebausteine, zu adressieren. Der Dritte im Bunde der wichtigsten Busse ist der **Kontrollbus**. Auf ihm werden die Kontrollsignale zur Selektierung und Überwachung der meisten Chips geführt.

Der RESET-Taster am Rechner (auf dem Blochschaltbild links Mitte) löst zum Beispiel ein solches Kontrollsignal aus. Durch diesen Impuls, der über einen sogenannten Timer-Baustein geleitet wird, wird verursacht, daß der Prozessor eine vollständige Initialisierung des Rechners durchführt.

Anhand des Blockschaltbildes wollen wir jetzt von links nach rechts die einzelnen Bausteine und ihr Zusammenwirken aufeinander kurz beschreiben, bevor wir auf sie im Laufe der nächsten Ausgaben ausführlicher eingehen werden.

Der Multi-Function-Peripherie Baustein 68901, kurz **MFP** genannt, ist für die serielle Schnittstelle des Typs RS 232 C mitverantwortlich. Außerdem fängt er sämtliche Interrupt-Signale auf.

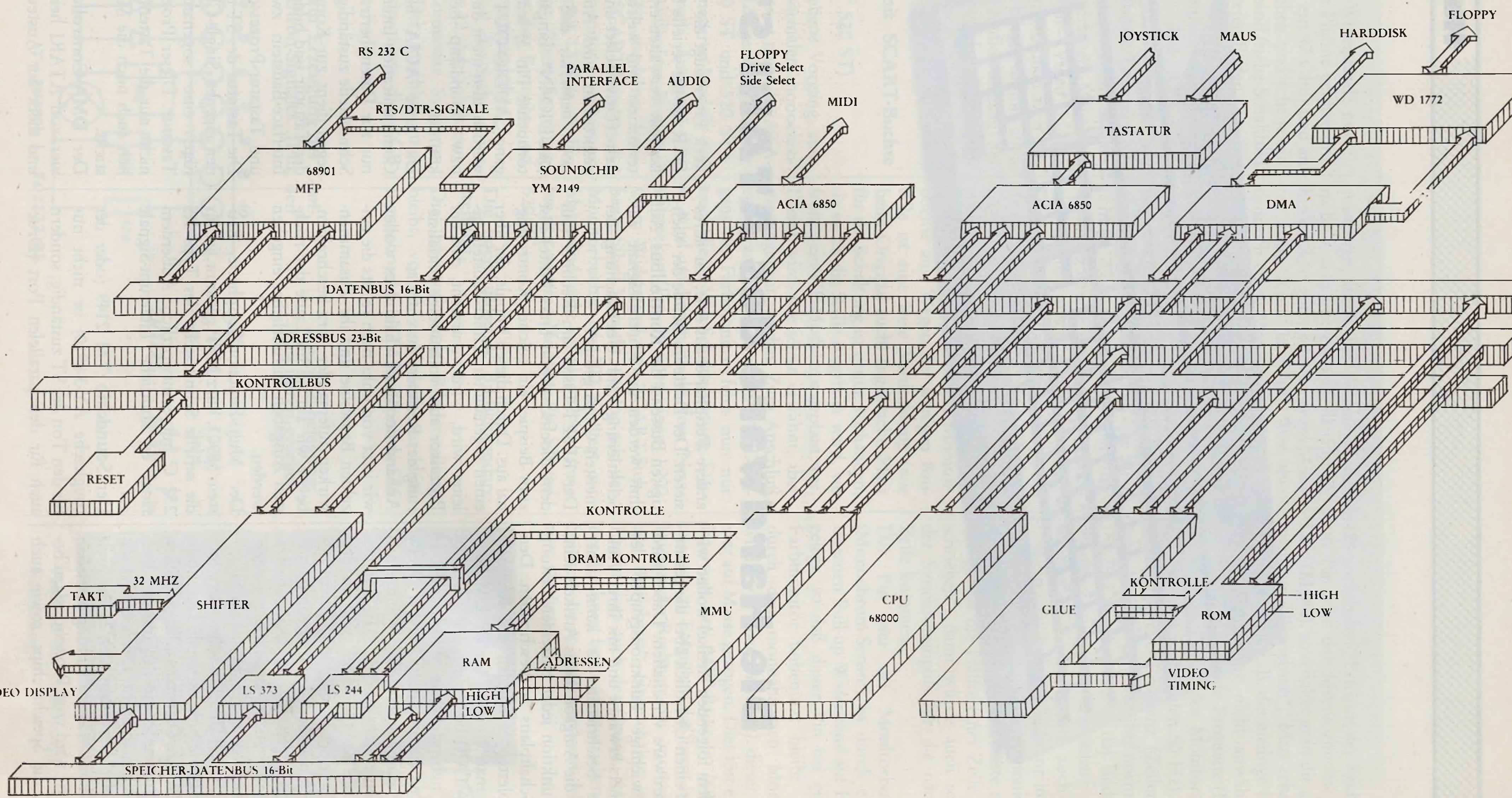
Der **Soundchip** YM 2149 (oder der baugleiche AY-3-8910) ist nicht nur für den Ton des ST zuständig, sondern auch für den parallelen Port (Drucker). Er erzeugt die Signale RTS und DTR für die serielle Schnittstelle. Außerdem verwaltet er, welche Diskettenstation und welche Seite der Diskette (Drive Select und Side Select) angesprochen wird. An dieser Stelle sei noch bemerkt, daß Tonsignale am „AUDIO-IN" Eingang (siehe Monitorbuchse Pin 5) lediglich zum Ausgang „AUDIO-OUT" hinzugemischt werden, also nicht durch den Rechner bzw. Soundchip bearbeitet werden können.

Die erste **ACIA** 6850 (Asynchron Communication Interface Adapter) ist nur zur Datenübertragung der MIDI-Schnittstelle zuständig. Die Zweite dagegen dient zur Kommunikation mit der Tastatur und bildet sozusagen den Interface-Baustein zwischen Rechner und Tastatur-Prozessor.

Die **Tastatur** des ST ist, da sie über einen eigenen Single-Chip-Rechner verfügt, eine sogenannte 'intelligente Tastatur. Dieser Prozessor überwacht nicht nur die Tasten, sondern zusätzlich auch noch die Maus und die Joysticks.

Der **DMA-Controller** ist einer der vier von ATARI hergestellten Chips und dient zur Ansteuerung für Flop-

Das Blockschaltbild des ATARI ST



pies, sowie einer Harddisk oder anderer externer Massenspeicher. Er ist übrigens, neben dem Video-Shifter, der einzige Baustein, der mit 16 Leitungen am Datenbus hängt.

Der **Floppy-Controller** WD 1772 wandelt die acht Bit breiten Daten des Rechners in serielle Daten für die Floppy um und umgekehrt. Außerdem liefert er alle für die Floppy notwendigen Steuersignale. ATARI hat damit eine Standardschnittstelle für Floppies eingebaut. Dadurch, daß der Controler frei programmierbar ist, wäre es denkbar verschiedene Formate zu betreiben.

Der System-**Haupttakt** von 32 MHz, im Blockschaltbild rechts unten, wird durch einen sehr einfach gehaltenen, 2-Transistor-Quarzoszillator, erzeugt . Diese hohe Frequenz wird dann für die verschiedensten Zwecke von anderen Bausteinen heruntergeteilt.

Der **Video-Shifter**, der ebenfalls von ATARI speziell für die ST-Serie entwickelt wurde, ist für den Bildaufbau des Rechners zuständig. Die Bildinformation holt er sich aus einem reservierten Speicherbereich aus dem RAM (32 KByte) und gibt sie auf dem Monitor aus. Die Farbinformation wird ebenfalls von ihm erzeugt. Der Shifter ist dafür über einen sogenannten Speicher-Datenbus mit dem RAM verbunden.

Die nächsten beiden Blöcke werden mit vier Standard-TTL-ICs (2x **74LS373** und 2x **74LS244**) realisiert. Sie stellen eine Verbindung des in High und Low geteilten RAM-Blocks über den Speicher-Datenbus zum System-Datenbus her. Kontrolliert werden sie dabei durch die MMU.

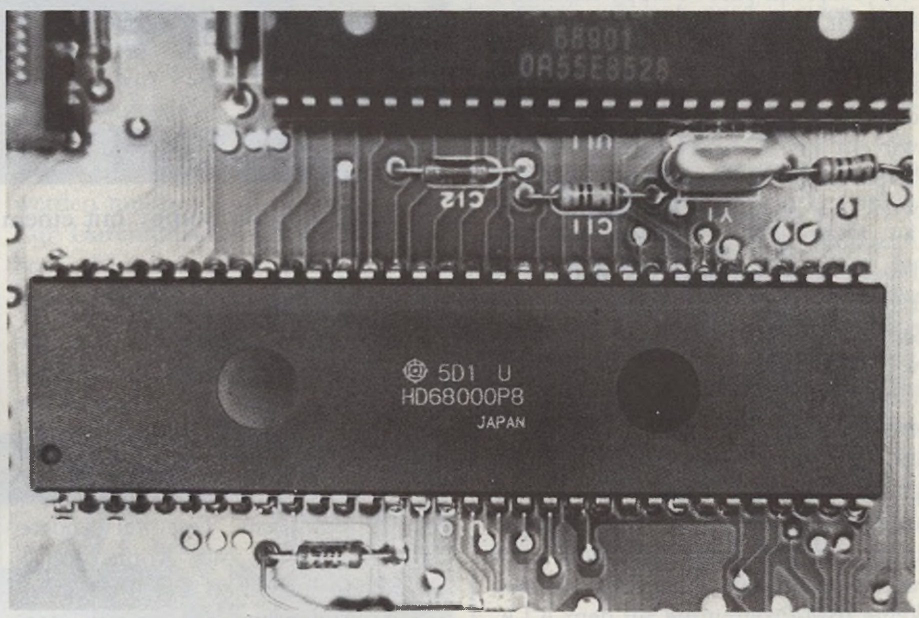
Der **RAM**-Speicher besteht aus dynamischen RAM-Chips, die jeweils 256 KBit speichern können. Der gesamte Speicher ist in eine High- und eine Low-Bank organisiert. Durch den MMU-Baustein werden die Speicher-Chips adressiert, sowie angesteuert. Je nach Modell ergibt sich ein Arbeitsspeicher von 512 Kilobyte bzw. 1 Megabyte.

Der nächste Baustein ist ein von ATARI entwickelter hochintegrierter Chip. Er heißt Memory Manager Unit (**MMU**), was soviel bedeutet wie Speicher-Verwaltungs-Einheit. Einer seiner Aufgaben besteht darin die Adressen für die RAMs zu multiplexen und den Bildschirmspeicher für den Video-Shifter zu selektieren. Eine andere wesentliche Aufgabe ist die Selektierung eines RAM Bereichs für den DMA-Betrieb.

Der Herz des ganzen Systems, ist der von Motorola entwickelte **68 000 Mikroprozessor**. Er wird als einer der leistungsfähigsten Prozessoren der neuen 16 Bit Generation bezeichnet. Das alles ist einerseits auf die hohe Taktfrequenz von 8 MHz und andererseits auf seine 16/32 Bit Struktur zurückzuführen. Durch die großzügig angelegte Registerzahl kann bei der Datenverarbeitung der Zugriff auf die externen RAM-Speicher eingeschränkt werden. Dies erhöht die Arbeitsgeschwindigkeit enorm. Ein weiterer Pluspunkt des 68 000 Prozessors ist die hohe Anzahl an Maschinenbefehle, die man intern implementiert hat. Das kommt dem Anwender hilfreich entgegen. Der Prozessor steuert und überwacht fast alle Abläufe im Rechner.

Der **GLUE**-Chip gehört auch zu den vier Bausteinen, die ATARI selbst entwickelte und herstellt. Von diesen hat der GLUE die vielfältigsten Aufgaben im Rechner zu erledigen. Er hat sozusagen überall seine „Hände“ im Spiel, so daß er seinen Namen GLUE, zu deutsch Leim oder Klebstoff, zu recht erhalten hat. Zu seinen Aufgaben gehört die Erzeugung fast aller Chip-Selekt-Signale (Dekodierung) von RAM, ROM und Peripherie-ICs. Ferner erzeugt er durch interne Teilerketten die Taktfrequenzen für die ACIAs und den Soundchip. Für den Monitor erzeugt er unter anderem die Synchronisations- und Austast-Signale. Außerdem erzeugt der GLUE Signale für die Interrupt-Steuerung, sowie für den DMA-Betrieb. Nicht zuletzt durch den Einsatz solcher hochintegrierter Spezial-Chips, die eine Unmenge herkömmlicher Standard-ICs ersetzen, verdankt der ATARI ST seinen günstigen Preis.

Der ATARI ST ist intern mit sechs **ROM**-Steckplätzen (nur Lese-Speicher) ausgestattet. Jeder dieser Steckplätze kann ROMs bis zu 32 Kilobyte aufnehmen, so daß sich maximal 192 Kilobyte ROM im Rechner unterbringen lassen. Auch dieser ROM-Speicher ist in eine Low- und eine High-Bank unterteilt. Ein 16-Bit Datenwort ist also in zwei verschiedenen ROMs abgelegt, die unteren 8 Bit im Rom der Low-Bank und die oberen acht entsprechend im ROM der High-Bank. Zur Zeit sind jedoch nur zwei dieser Steckplätze mit je acht Kilobyte ROMs bzw. EPROMs (2764) bestückt. In diesen Speichern befindet sich das Boot-Programm. Wenn Anfang 86 das TOS im ROM lieferbar ist, wird der verfügbare ROM-Speicher mit diesem gefüllt werden. Es sei noch bemerkt, daß man durch den Modulschacht weitere 128 Kilobytes an ROM-Speicher anschließen kann.



Anschluß eines 5 1/4 Zoll Laufwerkes an den ATARI ST

Der ATARI ST liest IBM-Daten

ATARI verwendet bei der ST-Serie unglücklicher Weise einige DIN Buchsen bzw. Stecker für die Verbindung der Außenwelt (Peripherie). Diese Verbindungen haben einen schlechteren elektrischen Kontakt, sowie einen mechanisch lockeren Halt (siehe Monitorstecker). Warum hat ATARI diese für einen Computer ungewöhnlichen Steckverbindungen verwendet? Wahrscheinlich spielt der Preis keine unbedeutende Rolle. Umso verwunderlicher ist es, daß man beim genaueren Ansehen der Floppy-Leitungen feststellt, daß sich hinter dieser fremden Steckverbindung ein standardmäßiger „Shugart-Bus" verbirgt. Durch diese Feststellung animiert probierten wir ein TEAC Laufwerk (5 1/4-Zoll) Typ FD-55 FV mit dem ATARI Laufwerk zu koppeln. Nachdem dieser Anschluß glückte, stellten wir mit Freude fest, daß sogar **IBM-PC** Disketten gelesen werden können. Man kann folglich Daten oder Text-Files vom IBM in den ATARI ST einlesen. Umgekehrt funktioniert dies selbstverständlich auch. Man kann z. B. Texte, die mit einem Textverarbeitungsprogramm auf dem ATARI ST erstellt und auf der 5 1/4-Zoll Floppy gespeichert wurden später in einem IBM weiterverarbeiten. Natürlich können keine IBM-Programme auf dem ATARI ST laufen, da wie bekannt sein dürfte im IBM ein anderer Mikroprozessor verwendet wird, und somit der Maschinencode nicht übereinstimmt. Mit Sicherheit ist es kein Zufall, daß ATARI dieses weitverbreitete Diskettenformat für seine ST Serie benutzt. Es bringt endlich mal eine Kompatibilität zu anderen Herstellern. Außerdem dürfte ein 5 1/4-Zoll Laufwerk in Verbindung mit dem CP/M 80 Emulator, der jetzt lieferbar sein müßte, sehr interessant sein. Zusätzlich hat es einen finanziellen Vorteil ein solches Laufwerk anzuschließen, denn einerseits sind die 5 1/4-Zoll Disketten preislich günstiger als die 3 1/2-Zoll

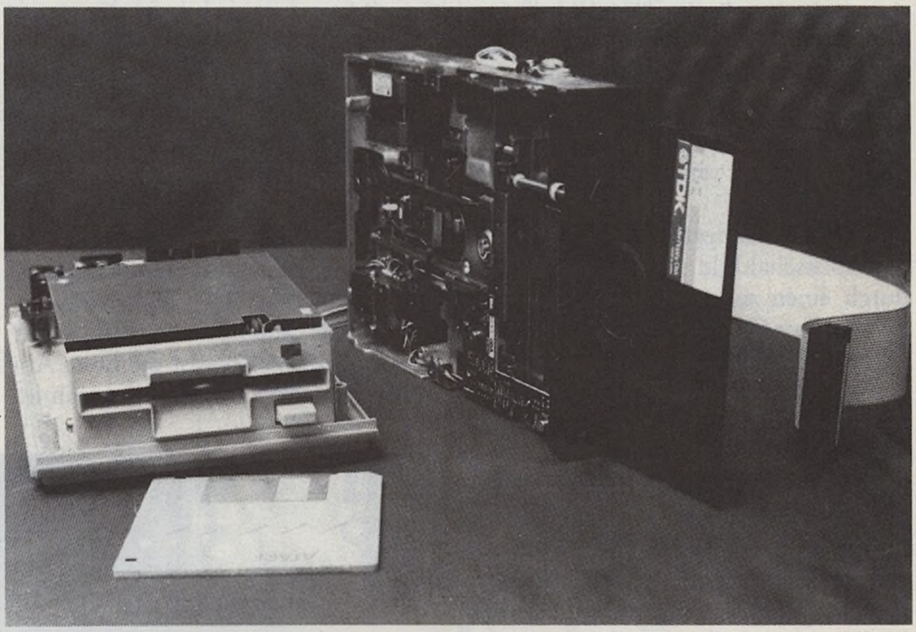


Bild 1: ATARI Floppy mit einem 5 1/4-Zoll Laufwerk gekoppelt

Disketten, andererseits kostet das verwendete TEAC Laufwerk, mit der gleichen Speicherkapazität wie die große ATARI Floppy SF 314, deutlich unter DM 500,–. Allerdings benötigt man dann noch ein Netzteil, das 12 Volt und 5 Volt für die Floppy liefert. Das bei unserem Test verwendete Laufwerk besitzt zwei Köpfe, sodaß 720 KByte formatiert zur Verfügung stehen. Außerdem läßt es sich zwischen 40 und 80 Tracks umschalten. Das bringt den Vorteil, ein sehr universelles Laufwerk zu besitzen. Selbstverständlich kann man auch andere 5 1/4 Zoll Floppies verwenden, unter der Voraussetzung, daß man ein paar technische Details seiner Floppy kennt.

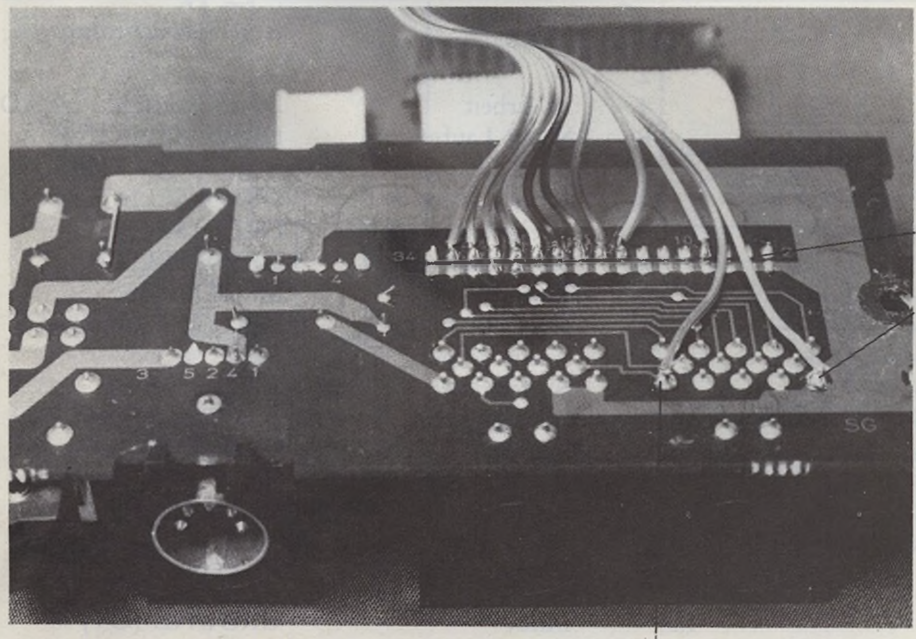
Bevor wir beschreiben, wie das 5 1/4-Zoll Laufwerk angeschlossen wird noch eine Warnung: Der Floppy-Anschluß sollte nicht von einem Laien, der sich noch nicht mit elektronischen Bastelarbeiten beschäftigt hat vorgenommen werden. Obwohl sich der eigentliche Anschluß nur auf ein 12-adriges Kabel bezieht, kann eine Fehlersuche oft langwierig und eventuell auch kostspielig sein. Da der 14-polige DIN-Stecker der Floppy, zur Zeit jedenfalls, noch nicht einzeln erhältlich ist, muß das vorhandene ATARI Laufwerk geöffnet werden und das neue Kabel von innen heraus geführt werden. Dabei kann man nach zwei verscheidenen Methoden vorgehen. Die nun folgend beschriebene Methode haben wir bei unserem Testmuster angewandt.

An Materialien ist dazu im Prinzip nur ein 12-adriges Kabel und ein 34poliger Platinenstecker, der auf die 5 1/4-Zoll Floppy gesteckt wird notwendig. Zuerst wird die ATARI Floppy geöffnet. Dazu sind die vier äußersten Schrauben, die sich auf dem Gehäuseboden befinden zu öffnen. Darunter befindet sich das eigentliche Floppy Laufwerk, welches durch ein Metallblech rundherum abgeschirmt ist, und die Anschlußplatine auf der deutlich zwei Spulen zu erkennen sind. Diese Platine ist über ein graues 34-poliges Flachbandkabel, das in der Pinbelegung dem Standartbus ent-

spricht, mit dem eigentlichen Laufwerk verbunden. Außerdem sind zusätzlich vier Leitungen für die Stromversorgung vorhanden (siehe Bild 1). Die soeben beschriebenen Kabel müssen vom Laufwerk abgezogen werden. Im folgenden Schritt ist das 12-adrige Kabel wie das Foto zwei und die Tabelle zeigen von unten an die Anschlußplatine anzulöten. Dabei ist darauf zu achten, daß das Signal „Drive Select 1" (Erkennungssignal für das neue Laufwerk) an die DIN Eingangsbuchse an Pin sechs angelötet wird. Da die meisten Laufwerke über vier Ansprechleitungen (Drive Select 0 bis Drive Select 3) verfügen, spielt es keine Rolle an welche die vorher beschriebene Leitung (vom Pin 6 der DIN Buchse) am neuen Laufwerk angeschlossen wird. Jedoch ist der Jumper (kleiner Verbindungsstecker oder Drahtbügel auf ihrer 5 1/4-Zoll Floppy Platine) auf die entsprechende Verbindung zu stecken. Das bedeutet: lötet man das von der ATARI Floppy kommende Drive Select 1 (DS1) Signal z. B. an Drive Select 0, der neuen Floppy an, so muß der Jumper (Brücke) auch auf Drive Select 0 (DSO) stecken. Alle anderen Kabel werden 1:1 verbunden. Dies bedeutet, daß eine Leitung von Pin acht kommend, auch wieder auf Pin acht der neuen Floppy geht. Es sei bemerkt, daß alle ungradzahligen Pins, folglich eine ganze Steckerseite, des 34-poligen Steckverbinders mit Masse verbunden ist. Also die Pins 1, 3, 5, ..., 33. Bevor wir die zweite Methode zur Herstellung der Verbindung beschreiben, muß folgendes erwähnt werden, was grundsätzlich zu beachten ist.

Die meisten Laufwerke (uns ist kein anderes bekannt) besitzen in der Eingangsstufe sogenannte Pull-up Widerstände. Diese sind oft in Widerstands Arrays zusammengefaßt. Sämtliche dieser Pull-up Widerstände müssen entfernt werden, da solche schon in der original ATARI Floppy enthalten sind. Entfernt man diese Widerstände nicht, überlastet man den Floppy-Controler im ATARI ST, was zur Folge hat, daß keines der Laufwerke mehr anspricht. Der Widerstandswert, solcher Pull-up Widerstände beträgt meistens zwischen 100 und 1 000 Ohm. Bei unserem FD-55 FV Laufwerk ist dieses Widerstands Array sogar in einer 14-poligen IC-Fassung steckbar gewesen, so daß es nur herausgenommen werden mußte.

Zur Durchführung der zweiten Methode benötigt man eine 34-polige Quetschverbindung, bestehend aus Buchse und Stecker (jeweils zum Quetschen). Die Buchse wird nun auf das kurze 34-adrige Kabel der ATARI Floppy gepreßt. Somit hat man sich einen passenden Anschluß für das neue Laufwerk geschaffen. Jetzt braucht man sich nur noch ein 34-poliges Verbindungskabel zu basteln, das auf der einen Seite, den zur oben genannten Buchse, entsprechenden Stecker erhält, und am anderen Ende des Flachbandkabels eine 34-polige Quetschbuchse, die auf die Platine der 5 1/4-Zoll Floppy paßt. Und schon haben wir unsere Signale an der zweiten Floppy! Das heißt eine Leitung fehlt noch. Es ist das Select Signal für das zweite Laufwerk (Drive Select 1). Diese Leitung hat ATARI von der DIN Eingangsbuchse (Pin 6) auf die DIN Ausgangsbuchse auf Pin 5 gelegt, sodaß ein dort angeschlossenes Laufwerk immer als zweites Laufwerk vom Rechner erkannt wird. Leider fehlt dieses Signal jedoch auf unserem 34-poligem Flachbandkabel. Folglich müssen wir nur noch eine Drahtbrücke von Pin sechs, der DIN Eingangsbuchse zum Anschluß 12 des Flachbandkabels auf der Platinenunterseite löten. Mit dieser Brücke auf Pin 12 der Anschlußleiste erfolgt die Selektierung des neuen Laufwerks immer durch Drive Select 1 (DS1). Folglich ist der Jumper im Zweitlaufwerk auf DS1 zu setzen. Diese Methode scheint uns die sicherste zu sein (man hat nur eine kleine Drahtbrücke zu löten). Hat man alle Verbindungen noch einmal kontrolliert, kann es zum ersten Probedurchlauf kommen.



Anschlußstecker (J2)

Masse

Pin 6 der DIN Eingangsbuchse (Drive Select 1)

Bild 2: Anschlußplatine der ATARI Floppy von unten gesehen

Nach dem Einschalten aller Geräte, sollte der Rechner das TOS von der ATARI Floppy booten. Meldet sich das Desktop in gewohnter Weise, so ist die erste Hürde überwunden. Klickt man nun das Laufwerk B an, müßte das neu angeschlossene Laufwerk anfangen zu arbeiten. Natürlich kann der Rechner die 5 1/4-Zoll Diskette, da sie noch nicht oder nicht im richtigen Format formatiert ist, nicht lesen. Es sollte folgende Fehlermel-

dung erscheinen: „Laufwerk B spricht nicht an, bitte überprüfen...". Als darauffolgender Schritt muß die Diskette formatiert werden. Erscheint kurz danach die Meldung, daß entweder ca. 360 bzw. ca. 720 KBytes freier Speicherplatz zur Verfügung steht, kann man beruhigt aufatmen, denn die Arbeit ist erfolgreich beendet. Unser TEAC Laufwerk läuft seit geraumer Zeit fehlerfrei. Erscheint jedoch die Fehlermeldung „Bitte Diskette B in Laufwerk A einlegen", so hat der Rechner beim booten des Betriebssystems das neue Laufwerk gar nicht anerkannt, somit muß also ein Fehler vorhanden sein. Hier müssen nochmals alle Verbindungen, Jumper und Pull-up Widerstände kontrolliert werden.
Zum Schluß sei noch bemerkt, daß natürlich auch 3 1/2-Zoll Laufwerke mit Shugart-Bus angeschlossen werden können. Ebenso spielt es keine Rolle, ob man ein Laufwerk mit nur einem oder eines mit zwei Köpfen benutzt. Es können jedoch nur maximal zwei Floppies angeschlossen werden. Generell sollte die Stepprate des Steppermotors kleiner oder gleich fünf Millisekunden betragen. Diese Forderung wird jedoch von fast allen modernen Laufwerken erfüllt. Es ist übrigens normal, daß die Antriebsmotoren beider Floppies anfangen zu laufen, wenn eine angesprochen wird.

34-poliges Flachbandkabel

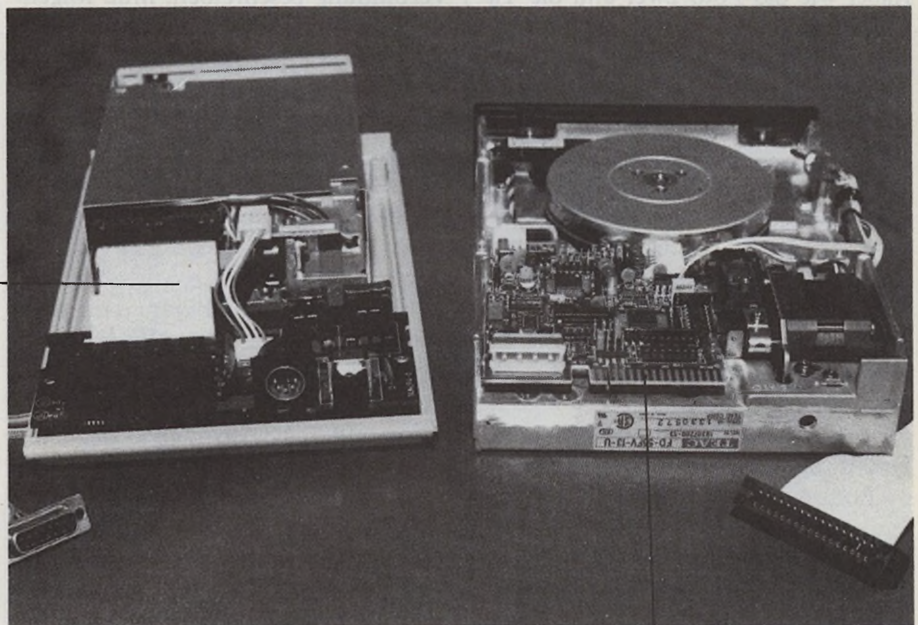


Bild 3: Rückansicht der Floppies

Platinenstecker (34-polig)
dahinter: 14-polige Fassung der Pull-up Widerstände

Belegung des 34-poligen Standard-Busses

ATARI Floppy (Epson)
Belegung des Anschlußsteckers (J2) auf der Anschlußplatine

5 1/4-Zoll Floppy
(z. B. FD-55 FV)
Platinenstecker

Signal	**Pin**		**Pin**	**Signal**	**engl. Bezeichnung**
—	2		2	—	—
—	4		4	In Arbeit	IN USE/HEAD LOAD
—	6		6	Selekt Laufwerk 3	DRIVE SELECT 3
Index Impuls	8	- - - - - - - - - -	8	Index/Sektor	INDEX/SECTOR
Selekt Laufwerk 0	10		10	Selekt Laufwerk 0	DRIVE SELECT 0
—	12	★- - - - - - - - -	12	Selekt Laufwerk 1	DRIVE SELECT 1
—	14		14	Selekt Laufwerk 2	DRIVE SELECT 2
Motor ein	16	- - - - - - - - - -	16	Motor ein	MOTOR ON
Richtung	18	- - - - - - - - - -	18	Selekt Richtung	DIRECTION SELECT
Schritt-Impuls	20	- - - - - - - - - -	20	Schritt-Impuls	STEP
Schreibdaten	22	- - - - - - - - - -	22	Schreibdaten	WRITE DATA
Schreibgatter	24	- - - - - - - - - -	24	Schreibgatter	WRITE GATE
Spur 00	26	- - - - - - - - - -	26	Spur 00	TRACK 00
Schreibschutz	28	- - - - - - - - - -	28	Schreibschutz	WRITE PROTECT
Lesedaten	30	- - - - - - - - - -	30	Lesedaten	READ DATA
Selekt Seite 0	32	- - - - - - - - - -	32	Selekt Seite 0	SIDE ONE SELECT
—	34		34	Fertig	READY
Masse	1,3,– 31,33	- - - - - - - - - -	1,3,– 31,33	Masse	GROUND (OV)

Die 12 gestrichelt dargestellten Leitungen sind die benötigten Signale.
★ von der DIN-Eingangsbuchse (Pin 6) kommend!

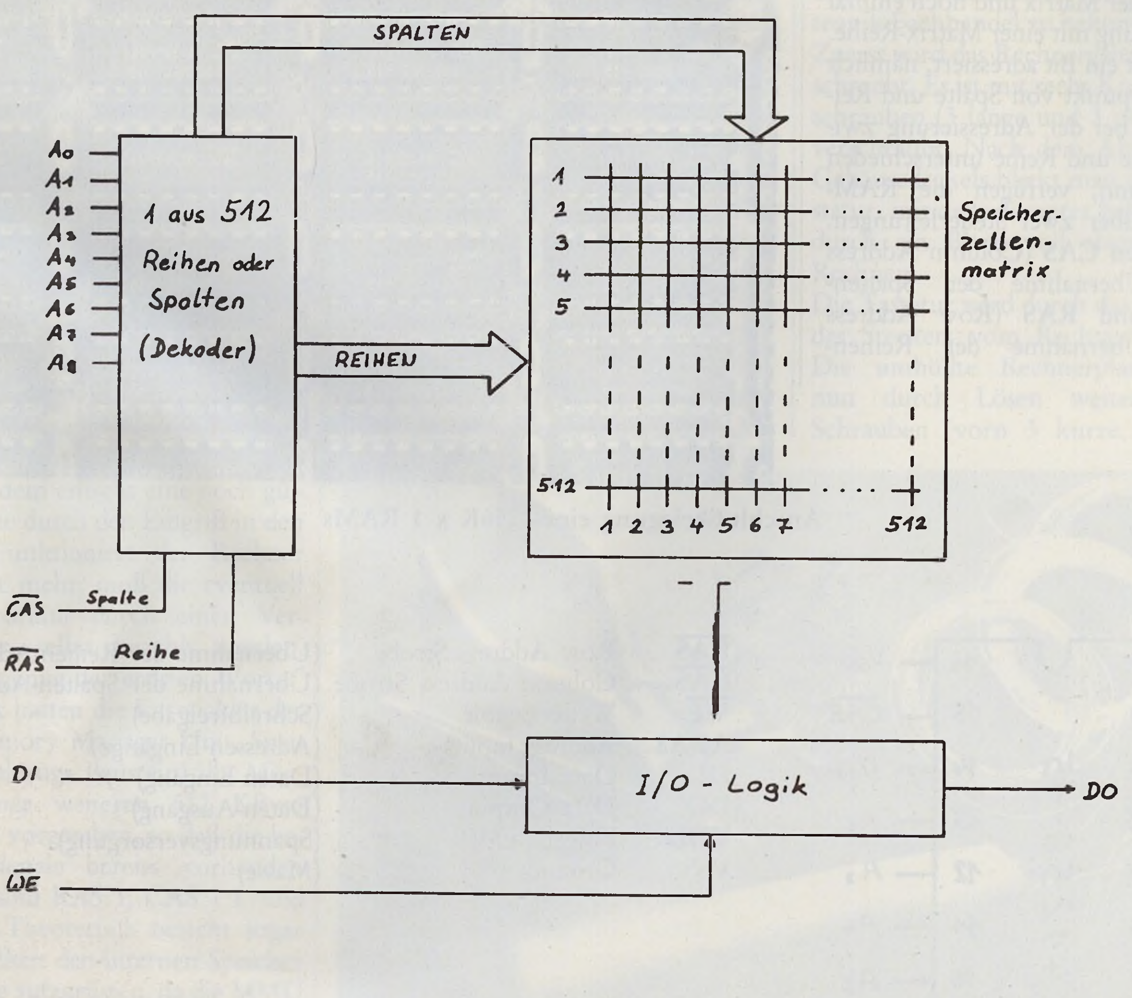
RAM-Bausteine

Im Hinblick auf den folgenden Artikel über die Aufstockung des „kleinen“ ATARI ST auf 1MByte RAM, erfolgen hier für den technisch interessierten Leser ein paar grundlegende Informationen über diese vieldiskutierten und wichtigen Speicherbausteine. Dabei wird auf die im ATARI ST verwendeten Bausteine besonders eingegangen. Das Verständnis dieses Abschnittes ist jedoch für den reinen Umbau des Rechners nicht unbedingt notwendig.

RAM (Random Access Memory, Speicher mit beliebigen Zugriff) ist ein Speicher, der seine Daten nur solange bewahrt, wie er mit Strom versorgt wird (im Gegensatz zum ROM). Grundsätzlich unterscheidet man zwei Arten, die statischen und die dynamischen RAMs. Die **statischen RAMs** speichern ihre Informationen in sogenannten Flipflops. Dadurch sind sie problemlos in der Anwendung, sie benötigen keinen refresh-Impuls, sind aber um einiges teurer als die dynamischen RAMs. Im ATARI ST werden, wie bei den meisten anderen Rechnern, die **dynamischen RAMs** verwendet. Diese Speicherelemente speichern die Informationen (Daten, Programme) mittels sehr kleiner Kapazitäten, die auf dem Chip integriert sind. Um den sich entladenden Speicherkondensator immer wieder aufzufrischen, müssen externe Auffrisch- (refresh-) Schaltungen vorhanden sein. Dieser refresh-Impuls muß ungefähr alle fünf Millisekunden erfolgen, sonst geht die Information im RAM verloren. Die kleinste Informationseinheit, ein Bit genannt, benötigt nun eine dieser winzigen Kapazitäten im RAM, um gespeichert werden zu können. Um nun eine große Bitmenge speichern zu können, werden die Speicherelemente (Kapazitäten) auf den Knotenpunkt eines Gitters gelegt. Man spricht von der sogenannten Speichermatrix. Ist im RAM jedes Speicherelement (Bit) einzeln anwählbar, wie beim ATARI ST, so spricht man vom bit-orientierten Speicher. Es gibt auch Speicherchips die vier bzw. acht Daten-Bits parallel (gleichzeitig) schreiben bzw. lesen können. Zu der letzt genannten Gruppe gehören vor allem die EPROMs, diese speziellen Speicherbausteine wollen wir hier jedoch noch nicht erklären.

Aufbau eines 256k x 1 RAMs

Im ATARI ST werden die modernen 256K (262 144 Bit!) Chips verwendet. Jeder dieser Bausteine kann also 256K x 1· Bit speichern. Da der Datenbus 16-Bit breit ist, werden mindestens 16 dieser Bausteine (für jede Date ein Chip) benötigt.



Organisation: 256K x 1 Bit = 2^{18} Bit
Zugriffszeit: 150 ns

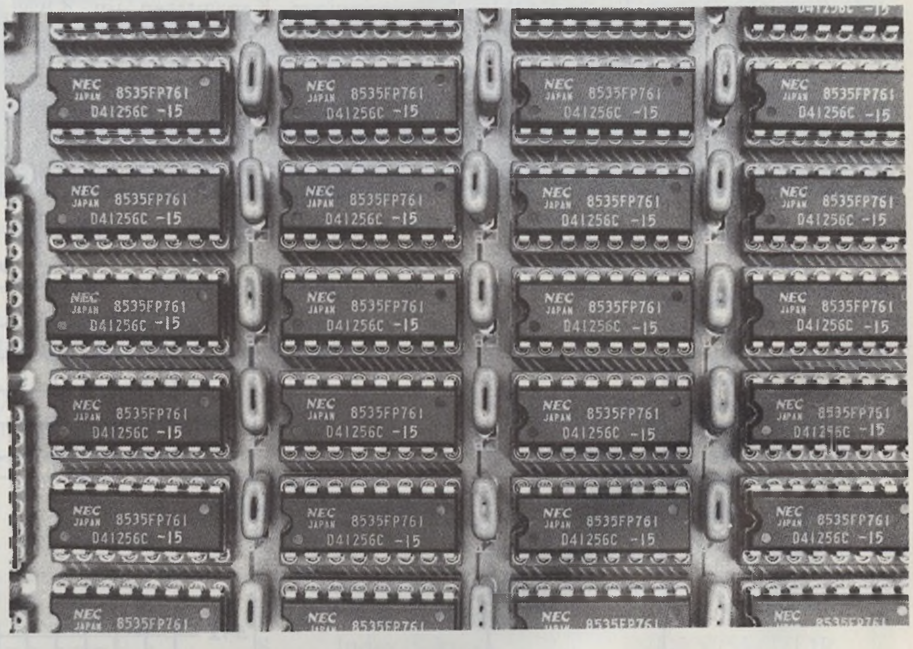
Aufbau eines 256K x 1 RAMs

Die Adressierung:

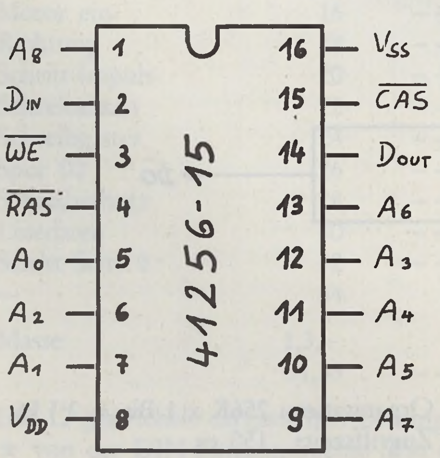
Um ein einzelnes Bit aus diesen 256K Bit anzusprechen, benötigt man neun Adreßleitungen, d. h. es müßten eigentlich 18 Adreßleitungen sein, da 2^{18} = 256K ergibt. Aus technischen Gründen und wegen der Gehäusegröße der Chips sind jedoch nur die Hälfte der Adreßleitungen, eben neun, herausgeführt worden (siehe Anschlußbelegung des RAMs). Um dennoch 256K adressieren zu können sind die Adreßleitungen der RAM-Bausteine „gemultiplext“. Was das bedeutet ist im folgenden erklärt.
Mit neun Adreßleitungen kann man eine Reihe bzw. eine Spalte aus einer Matrix von 2^9 = 512 auswählen. Eine 512-Matrix kann also 512 x 512 = 262 144 = 256 K Bit speichern (1K = 1 024). Zum Auslesen bzw. Beschreiben eines Bits müssen die neun Adreßleitungen folglich zweimal hintereinander (sequentieller Zugriff) aktiviert werden, einmal in Verbindung mit einer Spalte der Matrix und noch einmal in Verbindung mit einer Matrix-Reihe. Erst jetzt ist ein Bit adressiert, nämlich am Knotenpunkt von Spalte und Reihe. Damit bei der Adressierung zwischen Spalte und Reihe unterschieden werden kann, verfügen die RAM-Bausteine über zwei Steuerleitungen. Diese heißen **CAS** (Column Address Strobe, Übernahme der Spalten-Adresse) und **RAS** (Row Address Strobe, Übernahme der Reihen-Adresse).

Die Daten:

Wer sich die Anschlußbelegung angesehen hat, stellt fest, daß drei Pins (außer den zwei für die Spannungsversorgung) noch nicht erklärt wurden. Es sind die zwei Datenleitungen **DI** (Data Input, Daten Eingabe) und **DO** (Data Output, Daten Ausgabe), sowie das Schreib-Freigabe Signal **WE** (Write Enable). Wie der Name es schon sagt, kann das RAM nur beschrieben werden, wenn eine Date am Pin **DI** anliegt und der Freigabe Eingang **WE** einen LOW-Pegel (0-0,8BV) hat. Ist dagegen **WE** HIGH (2,4-5V), so kann aus dem RAM ein adressiertes Bit gelesen werden. Außerdem existiert ein besonderer Modus, bei dem **DI** mit **DO** verbunden wird. Durch ein geschicktes Timing ist es nun möglich über eine Leitung des RAM zu beschreiben bzw. zu lesen. Man hat also eine Zuleitung zu jedem RAM gespart. Von dieser Methode wird im ATARI ST Gebrauch gemacht.



Anschlußbelegung eines 256K x 1 RAMs



RAS	Row Address Strobe	(Übernahme der Reihen-Adresse)
CAS	Column Address Strobe	(Übernahme der Spalten-Adresse)
WE	Write Enable	(Schreibfreigabe)
A0-A8	Address Inputs	(Adressen-Eingänge)
DI	Data Input	(Daten-Eingang)
DO	Data Output	(Daten-Ausgang)
VDD	Power (+5V)	(Spannungsversorgung)
VSS	Ground	(Masse)

Ein Mega-Byte im Selbstbau

Vor weniger als zehn Jahren konnte man Heim-Computer kaufen, die gerade 1 Kilobyte freien Speicher hatten. Doch die Zeiten haben sich geändert. Immer größere und bessere Programme füllen die Speicher. Immer mehr Daten werden mit Rechner verarbeitet und immer neue Einsatzgebiete entstehen für die Computertechnik. Mit diesen gestiegenen Anforderungen an die Rechner wachsen natürlich auch die Anforderungen an die Speicher. Immer größere und schnellere Speicherchips sind die Folge, so daß zur Zeit 512 K- bzw. 1 024 KByte-Speicher den Stand der Technik bei Personal-Computern erfüllen. Doch kann man sicher sein, daß man in zehn Jahren, über die momentan so riesig erscheinenden Speicherkapazitäten, ein ähnliches Lächeln empfindet, wie wir heute über die 1 Kilobyte von damals.

Falls Sie nun einen 520 ST oder einen 260 ST besitzen und ihn auf ein Megabyte Arbeitsspeicher erweitern wollen, dann sollten sie die folgenden Zeilen aufmerksam lesen. Doch bevor wir die Aufrüstung zum Mega ATARI schrittweise erklären, eine Warnung: Alle anderen Umbau- oder Bastelarbeiten in diesem Heft sind im Vergleich zu der folgenden ein Kinderspiel. Sie sollten deshalb nur von erfahrenen Elektronikbastlern vorgenommen werden. Außerdem erlischt eine noch gültige Garantie durch den Eingriff in den Rechner. Funktioniert der Rechner später nicht mehr, muß die eventuell nötige Reparatur durch einen Vertragshändler, selbst bezahlt werden. Doch nun genug der ernsten Worte.

Zum Glück hatten die Entwickler der MMU (Memory Manager Unit, Speicher Verwaltungs Baustein) die Möglichkeit einer weiteren 512 KByte RAM-Bank vorgesehen, so daß die benötigten Signale bereits vorhanden sind. Dies sind RAS 1, CAS 1 L und CAS 1 H. Theoretisch besteht sogar die Möglichkeit den internen Speicher auf 4 MByte aufzurüsten, da die MMU ebenfalls eine zehnte Adressleitung zur Adressierung von 1 MBit-Chips bereit hält. Diese Chips sind zur Zeit jedoch kaum erhältlich und falls doch, zu kaum bezahlbaren Preisen.

Der 68 000 Prozessor verfügt über 16 Datenleitungen, folglich ist für jede Date ein RAM-Chip vorhanden. Bei den verwendeten 256K RAMs ergibt das eine Speicherkapazität von 512 Kilobyte. Um diese Kapazität auf 1 Megabyte zu erhöhen sind folglich 16 weitere RAMs erforderlich. Es ist darauf zu achten, daß möglichst nur RAM-Chips des Originals verwendet werden. Als problemlos erwiesen sich dynamische RAMs der Firma NEC mit der Typenbezeichnung D41256C-15. Schwierigkeiten soll es bei der Verwendung von Bausteinen der Firma Texas Instruments (TI) gegeben haben. Bei unseren Tests verwendeten wir neben NEC auch Bausteine von Fujitsu. Wichtig ist, daß bei der Aufrüstung nur RAMs des gleichen Typs verwendet werden. Außerdem ist darauf zu achten, daß die Zugriffszeit der RAMs nicht größer als 150 Nanosekunden (ns) ist, langsamere Chips funktionieren nicht. Der Preis für solche RAM-Bausteine liegt zur Zeit zwischen DM 8,– und DM 15,– pro Stück.

An Werkzeug benötigen Sie eine kleine Flachzange, einen Seitenschneider, eine Pinzette und am Besten eine Lötstation mit temperaturgeregeltem Lötkolben. Dieser sollte eine feine Spitze besitzen, ein globiger 100 Watt Lötkolben ist hier völlig falsch. Außerdem brauchen Sie noch etwas dünnen Lötdraht und ca. 50 cm Schaltlitze bzw. versilberten Spulendraht mit mindestens 0,5 mm² Querschnitt. All diese Utensilien sollten im guten Elektronik-Fachhandel zu bekommen sein.

Zuerst wird das Rechnergehäuse aufgeschraubt. Es ist mit sechs Kreuzschlitzschrauben (3 lange und 3 mittellange) verschraubt. Nach dem Abheben des Gehäusedeckels blickt man auf die Tastatur, mit dem darunter befindlichen, durch ein Blech voll abgeschirmten Rechner.

Die Tastatur wird durch das Abziehen des Steckers vom Rechner getrennt. Die umhüllte Rechnerplatine wird nun durch Lösen weiterer sechs Schrauben (vorn 3 kurze, hinten 3

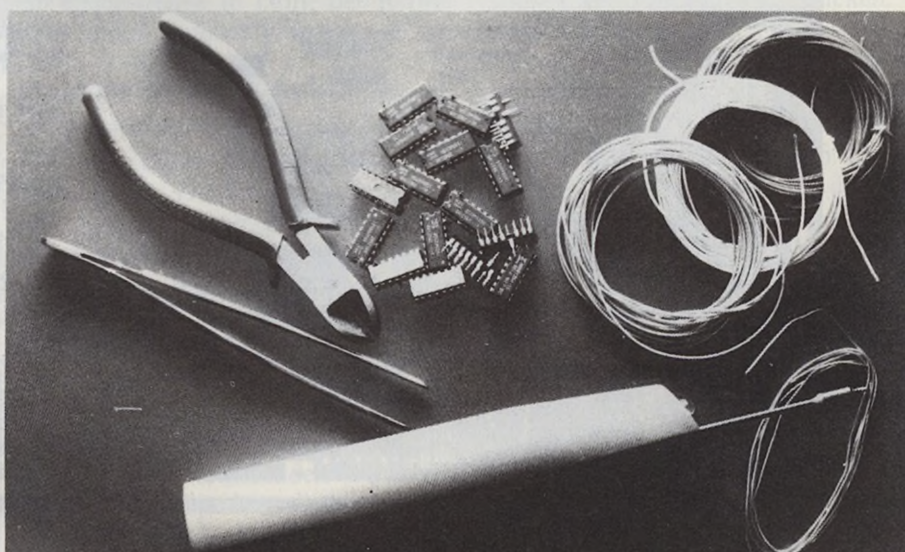


Bild 1: Das Werkzeug einschließlich der RAMs

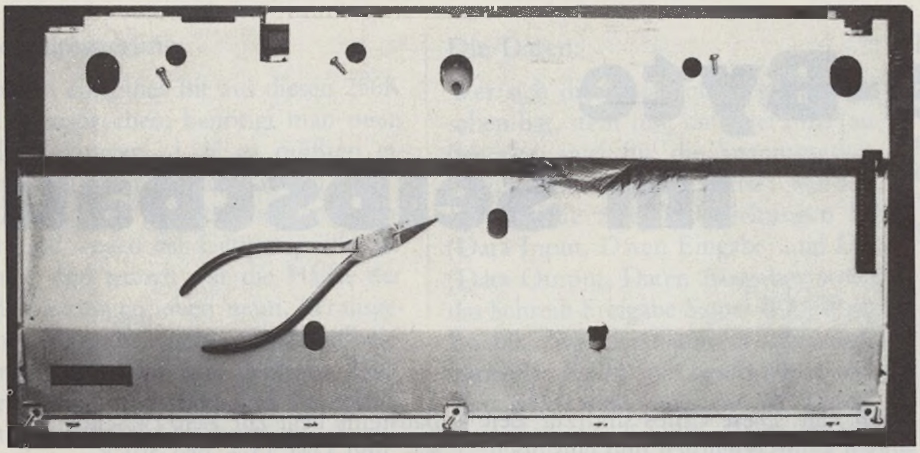


Bild 2: Rechner mit Abschirmblech

mittellange) von der Gehäuseunterhälfte gelöst (siehe Foto). Als nächstes müssen die 11 Blechbügel, die rund um die Platine verteilt sind, mit einer Flachzange geradegebogen werden. Leider sind bei einigen Rechner-Modellen zwei dieser Blechbügel mit dem Gehäuse verlötet. Diese Verbindung muß gelöst werden und braucht später nicht wieder verlötet zu werden. Jetzt kann das obere Abschirmblech nach hinten weggeklappt werden, und die eigentliche Rechnerplatine ihrer Metallschale entnommen werden.

Nun müssen die neuen RAM-Chips auf ihren „Arbeitsplatz" vorbereitet werden (siehe Foto). Dazu werden die Pins 4 und 15 aller RAMs nach oben gebogen (Pin eins befindet sich links von der Einkerbung, wenn man von oben auf die Chips schaut). Jetzt können sie auf die vorhandenen RAMs des Rechners aufgesteckt werden, dabei ist darauf zu achten, daß die Einkerbungen der neuen mit den im Rechner befindlichen übereinstimmen (sie zeigen zur Platinenmitte).

An dieser Stelle sei bemerkt, daß es schon mindestens drei verschiedene Rechnerplatinen gibt, von denen jedoch nur die erste Version deutlich von den anderen abweicht. Bei dieser ersten Version sind fast alle ICs (außer dem Prozessor und einem Optokoppler) gesockelt. Außerdem ist der Videoteil und der Systemoszillator nicht abgeschirmt. Hat man einen solchen Rechner vo sich, ist das Festlöten der „huckepack"-sitzenden RAMs einfacher, dafür gibt es Schwierigkeiten beim Zusammenbau mit dem oberen Abschirmblech. Entweder lassen Sie es ganz weg, oder Sie sind ein Feinmechaniker und sägen, für die zu hohen RAMs, einen schmalen Spalt in das Blech. Da nur ca. 750 Stück dieser ersten Version verkauft wurden, ist die Chance einen solchen Rechner zu haben relativ gering. Bei allen anderen Modellen sind die ICs, außer der vier von ATARI entwickelten, direkt auf die Platine gelötet.

Kommen wir jetzt zum schwierigsten Teil der gesamten Umbau-Aktion. Die 16 zusätzlichen RAMs müssen mit den darunterliegenden verlötet werden. Natürlich bleiben die zwei nach oben abgebogenen Beinchen noch frei, sie werden danach separat verdrahtet. Bitte lassen sie sich für diese Arbeit Zeit und löten mit einer ruhigen Hand ein Pin nach dem anderen fest. Unsere Erfahrung zeigt, daß man oft aus übertriebener Vorsicht (Zerstörung durch Überhitzung) den Lötkolben eher zu kurz als zu lang an eine Lötstelle hält. Schlechte und kalte Lötverbindungen sind die Folge. Schon eine schlechte Verbindung genügt, und der Rechner erkennt die zusätzliche RAM-Bank nicht an.

Nach dieser diffizilen Arbeit sind nun die noch freie, nach oben gebogenen Beinchen an der Reihe. Der Pin vier aller 16 RAMs wird mit der oben erwähnten, abisolierten Schaltlitze verbunden bzw. verlötet. Da jede RAM-Bank in ein High- und ein Low-Byte unterteilt ist, wird der Pin 15 von jeweils acht RAMs verbunden (siehe Fotos).

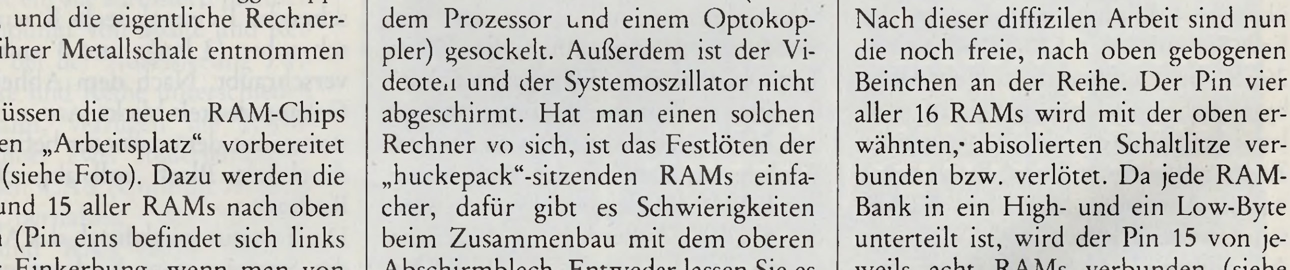


Bild 3: 520 ST+

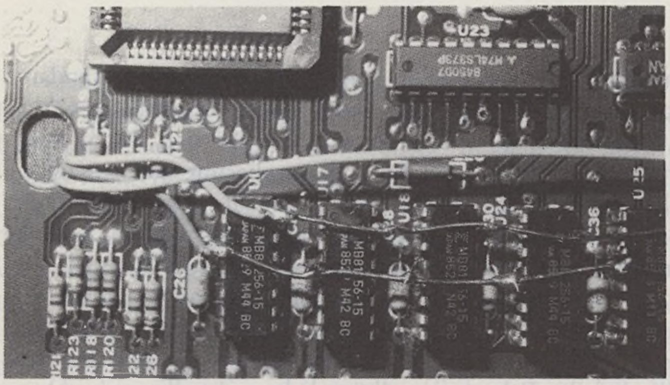


Bild 4: (CAS 1L)

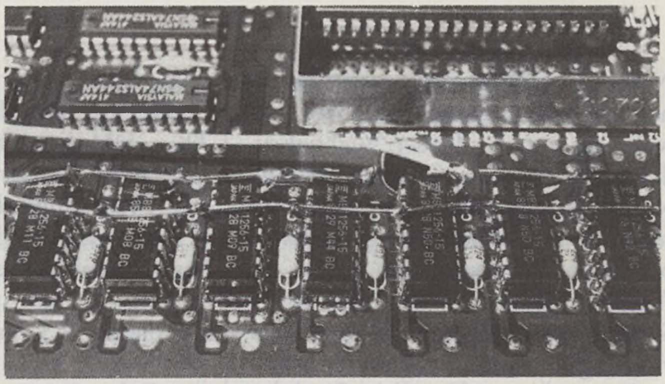


Bild 5: (CAS 1H)

Der Pin 15 der **linken** acht RAMs wird nun über eine isolierte Leitung zur MMU Pin 21 (CAS 1 L) geführt, und der gleiche Pin der **rechten** acht RAMs zum Pin 22 (CAS 1 H) der MMU. Der Pin 18 (RAS 1) der MMU wird über eine Leitung mit Pin vier aller RAMs verlötet. Die drei isolierten Zuleitungen werden von der MMU an der Unterseite der Platine nach oben auf die RAMs geführt, und zwar am Besten durch das Loch, das sich zwischen der MMU und dem GLUE-Chip befindet. Dies sind die beiden quadratischen Chips auf der Platine (siehe Foto).

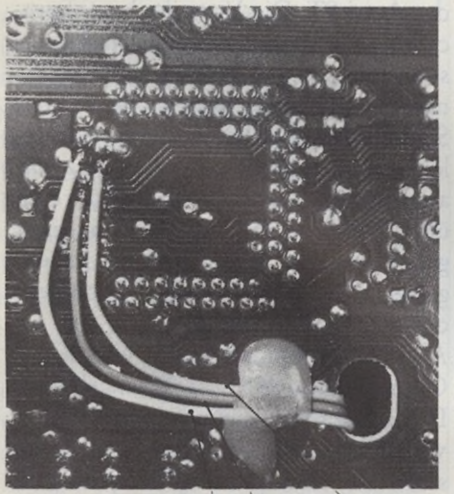


Bild 6: MMU

Damit ist der Einbau der neuen RAM-Bank abgeschlossen. Jetzt müssen alle Verbindungen noch einmal genauestens überprüft werden. Zum Testen legt man die Rechnerplatine (ohne Abschirmung) am Besten in die untere Gehäusehälfte und verbindet sie mit den Steckern von Floppy, Monitor, Netzteil, Maus und Tastatur. Nach dem Einschalten muß der Rechner das TOS von der Diskette laden. Nun gibt es zwei Möglichkeiten, um festzustellen, ob der Rechner beim Initialisieren die neue Speicher-Bank anerkannt hat oder nicht.

1.) Nach dem Laden des mitglieferten BASICs wird im Direktmodus PRINT FRE (0) eingegeben. Darauf sollte im OUTPUT-Fenster ein Wert größer als 520 000 erscheinen. Die Angabe eines genauen Zahlenwerts ist wegen der verschiedenen BASIC-Versionen nicht möglich.

2.) Beim Kopieren einer ganzen Diskette mit nur einer Floppy sollte der eigentliche Kopiervorgang in einem Arbeitsgang erfolgen, d. h. ohne in der Mitte die Disketten wechseln zu müssen.

Hat der Rechner, wie wir hoffen, die neuen Speicher anerkannt, so kann er wieder zusammengebaut werden. Zuvor sollte jedoch, um Kurzschlüsse zu vermeiden, am oberen Abschirmblech ein Klebestreifen in Höhe der RAMs (von innen) angebracht werden.

Probleme nach dem Aufrüsten:

Normalerweise sollte es bei genauer Beachtung dieser Anleitung keine Probleme geben. Doch leider hatten auch wir, besonders mit einem Rechner, von sechs Umgebauten, Schwierigkeiten. Trotz sorgfältiger Kontrolle aller Verbindungen und Leitungen wollte dieser Rechner die neuen Bausteine nicht akzeptieren. Ein mehr oder weniger verzweifelter Versuch den GLUE-Chip, mit einem schon aufgerüsteten funktionierenden Rechner, zu tauschen brachte dann den erwünschten Erfolg. Bemerkenswert dabei ist die Tatsache, daß der „Tauschgeber" mit dem GLUE-Chip des nicht funktionierenden Rechners weiterhin einwandfrei arbeitet. Wie wir in Erfahrung bringen konnten, ist das Timing zwischen GLUE-Chip und VIDEO-SHIFTER teilweise kritisch, so daß es, durch Bauteiltoleranzen bei der Herstellung dieser Chips, eventuell zu der oben genannten Störung kommen kann. Den GLUE-Chip kann man übrigens mit zwei kleinen Schraubendrehern oder abgewinkelten Pinzetten aus seiner Fassung hebeln. Dazu steckt man diese in die beiden Schlitze der Fassung und hebelt nun mit etwas Druck den Chip heraus (möglichst ohne die Fassung zu zerstören).

Eine weitere Möglichkeit für Störungen betrifft die Spannungsversorgung aller RAMs, diese ist nämlich über eine Drossel, zur nochmaligen Siebung der Spannung, geführt. Die Drossel befindet sich in der rechten unteren Ecke der Rechnerplatine. Sollte die Spannung an den RAMs (Pin 8 und 16) deutlich unter vier Volt absinken, so ist die Drossel mit einem Widerstand von einigen Ohm zu überbrücken.

Zum Schluß noch eine Bemerkung: Die neue 512 K RAM-Bank liegt in dem Adreßbereich von Hex $80 000 bis $FFFFF. Wer ein Monitor-Programm besitzt (hier ist kein Datensichtgerät gemeint), wie z. B. den SID aus dem Entwicklungspaket, der kann mit dessen Hilfe einen Speicherbereich füllen oder sich den Inhalt einzelner Speicherzellen ansehen. Somit hat man noch eine andere Möglichkeit die Funktion der RAMs zu testen und eventuelle defekte Zellen zu finden.

Wir hoffen nun alles getan zu haben, um den Umbau so leicht wie möglich zu gestalten.

Software für den Atari 520 ST

Neben den Hardwareeigenschaften eines Computers, ist die verfügbare Software von entscheidender Wichtigkeit, denn die meisten Computerbesitzer woillen, anstatt selbst als Programmierer aktiv zu werden, fertige Software benutzen. Bei dem Atari ST ist das Angebot der im Laden käuflichen Software momentan noch nicht sehr vielfältig. Viele namhafte Softwarehäuser haben aber umfangreiche Produkte für den Atari St angekündigt, stecken aber noch in der Entwicklung. Als bestes Beispiel sei die Firma Atari selbst genannt. Der größte Teil ihrer Software ist noch in der Entwicklung und ist zur Zeit nur in Vorabversionen erhältlich. Darunter fällt neben dem wichtigsten Produkt, dem TOS auch GEM Write und GEM Draw. Bei diesen Produkten ist es auch angebracht dem Kunden eine noch nicht voll funktionstüchtige Version zur Verfügung zu stellen, aber bei Softwareherstellern die ihre Produkte teuer verkaufen, ist es als selbstverständlich zu betrachten, daß sie ihre Produkte erst voll ausgereift auf den Markt bringen. So entsteht ein ziemlich großer Zeitdruck, denn alle Firmen wollen ihre Produkte möglichst schnell auf den Markt bringen, vor allem aber vor der Konkurrenz.

Bei dem Atari ST ist seine Prozessorverwandtschaft mit dem Apple Macintosh von Vorteil, denn dadurch ist es relativ leicht Software anzupassen. Die Firma FOCUS aus Hannover bietet eine Vielzahl solcher Programme an. Dazu gehören die gebräuchlichsten Programmiersprachen: Pascal, Basic und Fortran 77.

Bei dem Pascal-Compiler handelt es sich um ein echtes USCD-Pascal, das den vollen Pascalstandard nach Wirth und Jensen bietet. Desweiteren verfügt der Compiler noch über vielseitige Befehlserweiterungen zur Stringverarbeitung, 36-stelliger Arithmetikoperationen und Schnittstellen zu Assembler und GEM. Dadurch ist es möglich GEM-Routinen (Maus, Grafik, Windowtechnik etc.) in das Programm einzubinden. Preis: 890,– DM.

Der Fortran 77 Compiler entspricht dem ANSI 77 Standard und unterstützt neben grafischen Routinen auch komfortable Dateiverwaltungsfunktionen (Preis: 990,– DM).

Die dritte Programmiersprache im Bunde ist der Basic-Compiler für 790,– DM. Er enthält vor allem strukturierte und Zeilennummernunabhängige Programmierbefehle sowie Möglichkeiten zur Funktionserstellung mit Parameterübergabe ähnlich wie in Pascal und Fortran.

Zu diesen 3 Compilern gibt es ein Tool-Kit. Er enthält unter anderem einen Debugger, zur Hilfe bei der Fehlersuche, einen Maschinencode Generator sowie einen 68 000 Assembler (Preis 490,– DM).

Grundlage dieser Sprachen ist ein UCSD p-System, welches für alle Sprachen gleichermaßen aufgebaut ist. Es erzeugt einen Zwischencode, p-Code genannt, der auf sämtliche Computer, die dieses USCD p-System beherrschen, übertragbar ist. Somit sind Programme, die auf einem IBM, Wang, Apple, HP, DEC, NEC, TI oder einem von über 100 Computern, auf denen das p-System lauffähig ist, erstellt wurden, auf dem Atari 520 betriebsfähig. Mit diesen Programmen, die übrigens sofort lieferbar sind, erhält man einen Bildschirmeditor, der nützliche Utilitys, zum Beispiel: Text verschieben, suchen und ersetzen enthält. Darüber hinaus verfügt der Editor über die Möglichkeit Textauszüge abzuspeichern und zu laden sowie über komfortable Formatierungsmöglichkeiten, ähnlich einem Textverarbeitungsprogramm. Durch den einheitlichen p-Code ist es möglich Progarmme, die in verschiedenen Programmiersprachen erstellt wurden, zusammenzufügen.

Ein weiterer Lieferant für hochwertige Software ist der englische Hersteller METACOMCO. Er bietet neben einem Pascal Compiler und einem Macro Assembler noch einen C-Compiler, LISP Interpreter sowie einen BCPL Compiler an. Der Pascal Compiler arbeitet nach dem ISO Standard und erzeugt direkt verarbeitbaren Maschinencode. Er soll sich durch seine schnelle Compilierungszeit auszeichnen und durch die Übertragbarkeit sämtlicher Pascalprogrammen die dem ISO-Standard entsprechen (Preis 475,–).

Als weitere Programmiersprache bietet Metacomco einen LISP Interpreter. LISP ist eine Sprache aus dem Bereich der 'künstlichen Intelligenz'. Dem Programmierer bietet sie eine völlig neue Art der Programmerstellung, die stark von den üblichen Sprachen abweicht. Es handelt sich hier um den Einstieg in eine völlig neuartige Programmierweise, die in Zukunft sicherlich breitere Anwendung finden wird (Preis 795,–).

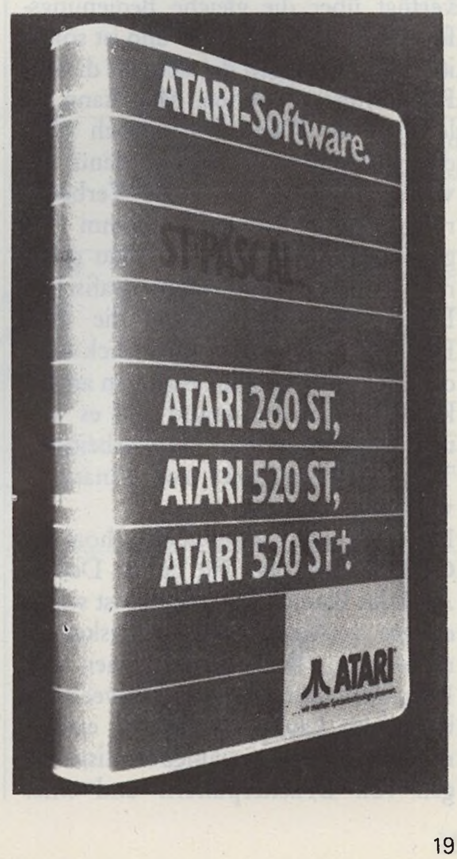
Die Programmiersprache 'C' kann man mit dem 'LATTICE C Compiler' für 595,– DM käuflich erwerben. Der Compiler soll über eine große Bibliothek verfügen, die mit der UNIX-Bibliothek übereinstimmt. C ist eine Sprache die heute schon starken Nutzen findet, zum Beispiel in mehreren Computerbetriebssystemen die vollständig oder teilweise in dieser Sprache geschrieben sind. Dazu gehört auch der Atari ST. Die Vorteile dieser Sprache liegen in ihrer universellen Anpassungsfähigkeit, die es dem Programmierer erlauben seinen benötigten Sprachumfang selbst zusammenzustellen. C ist strukturiert und trotzdem assemblernah und bietet somit eine überschaubare Programmierung und gleichzeitig die Eigenschaft der Erzeugung eines kompakten Maschinencodes.

Als weitere Sprache wird ein BCPL Compiler für 445,– DM angeboten. Diese Sprache zeichnet sich ähnlich wie C, durch die Erzeugung von schnellen und speicherplatzsparenden Programmen aus. Der Compiler bietet die Zugriffsmöglichkeit auf eine Vielzahl von GEM-Routinen, mit denen es zum Beispiel ermöglicht wird, die Grafik- und Window-Funktionen in. das BCPL-Programm zu übernehmen.

Zu guter letzt gibt es noch einen 68 000 Makro-Assembler zum Preis von 395,– DM, der schon lieferbar ist. Die übrigen vier Produkte von Metacomco sollen noch im Dezember erhältlich sein.

Modula 2 ist jetzt auch für den Atari

ST verfügbar. Die neue Sprache des Pascal-Entwicklers Wirth bietet alle Fähigkeiten von Pascal und darüber hinaus mehrere Verbesserungen. Modula 2 ist zur intensiven Nutzung der GEM Routinen ausgelegt. Dadurch wird weniger Speicherplatz benötigt, und eine kürzere Programmlaufzeit erreicht. Bei der Verarbeitung von größeren Datenmengen oder bei der Darstellung von bewegten Grafiken kommt dies besonders zur Geltung. 'Modula 2/ST', von TDI entwickelt, ist für 890,– DM sofort erhältlich.
Ein weiterer Fortran 77 Compiler kommt von der englischen Firma Prospero unter dem Namen 'Pro Fortran 77'. Er bietet den gesamten ANSI Standard und dadurch volle Übertragbarkeit mit anderen Rechnern, vor allem mit 68 000 Rechnern (MacIntosh, Amiga). Es handelt sich hier um einen 2-Pass Compiler der relokatiblen, also Speicherplatzunabhängigen, Maschinencode erzeugt. Es besteht die Wahlmöglichkeit zwischen geschwindigkeits- und speicherplatzoptimierter Kompilierung. Zusätzlich bestehen verschiedene Möglichkeiten zur Ausgabe von wichtigen Speicheradressen und Laufzeitangaben. Pro Fortran 77 verfügt über eine große Anzahl von Routinen in einer Run-time Bibliothek, die besonders auf schnelle Ausführzeit und Rechengenauigkeit zugeschnitten sind. Pro Fortran 77 wird für 900,– DM ab Ende Dezember auf dem Markt erscheinen.
Ein weiterer Hersteller ist die englische Firma KUMA. Sie bietet einen sehr kompakten Makro-Assembler, 'SEKA' an. Dieser Assembler erzeugt direkt ausführbaren Maschinencode mit einer extrem schnellen Geschwindigkeit von 30 000 Zeilen pro Minute. Es kann absoluter oder verschiebbarer Maschinencode generiert werden, der nicht erst, wie bei manchen anderen Assemblern, einem 'Linkprozess' unterworfen wird um lauffähig zu werden. Der 'SEKA' dürfte für denjenigen der möglichst einfach und schnell Assemblerprogramme schreiben will, das momentan günstigste Programm sein. Es belegt nur 17Kbyte Speicherplatz und ist somit auch für resistente Programmierung geeignet, d. h. man kann sein Programm im RAM kompilieren und von dort aus auch ausführen. Zum Lieferumfang gehört neben einem Bildschirmeditor auch ein Disassembler, ein Linker zum Zusammenbinden von mehreren Assemblerprogrammen sowie ein Maschinenmonitor. Dieses Programm ist unter anderem bei SOFTLINE zum Preis von 198,– DM erhältlich.
Ebenfalls bei dieser Bezugsadresse ist ein Forth Compiler zu beziehen (Preis: 498,– DM).
Ein weiterer englischer Softwarehersteller ist GST aus Cambridge. Er bietet einen C-Compiler und einen Makro Assembler an. Beide Produkte enthalten neben den eigentlichen Bestandteilen einen Bildschirmeditor, mit GEM bezogenen Editiereigenschaften, umfangreiche Bibliothekszugriffsmöglichkeiten sowie Schnittstellen zu GEM und TOS. Der Preis liegt bei dem C-Compiler bei etwa 80 Pfund und bei dem Makro Assembler bei 40 Pfund.
Neben Programmiersprachen existieren auch schon einige Anwenderprogramme. Neben dem GEM-Write, das noch nicht in der endgültigen Version erhältlich ist, gibt es schon mehrere Textverarbeitungsprogramme für den Atari ST. Zum ersten das SM-Text 520, von der Münchner Firma SM entwickelt, welches nun von Atari vertrieben wird und das C-Text, ebenfalls von einer deutschen Firma (C-Soft). Beide Programme waren bei Redaktionsschluß noch mit einigen kleinen Fehlern behaftet, die bei falscher Bedienung den Rechner zur 'Pilzproduktion' anregten. Aber kein Programm ist bekanntlich fehlerfrei, was sich manchmal erst nach vielen Betriebsjahren herausstellt.
C-Text wird zusammen mit C-Adress im Paketpreis von 250,– DM geliefert. Es handelt sich hierbei um ein Wordstar ähnliches Textverarbeitungssystem, das in Kombination mit C-Adress zum komfortablen Erstellen von Serienbriefen genutzt werden kann. Durch den niedrigen Preis eignet es sich besonders für diejenigen die nach dem Kauf des Rechners billigere Software bevorzugen. Näheres zu diesem Programmpaket sowie zu SM-Text 520 können Sie in den gesonderten Testberichten dieser Zeitschrift nachlesen.
Neben dem Programmpaket C-Text & C-Adress bietet C-Soft noch ein Programm zur Auftragsabwicklung, unter dem Namen 'C-Auftrag' an. Hiermit können Kundendateien und Lagerdateien verwaltet, Aufträge erstellt und Rechnungen geschrieben werden. Die Masken zur Datenaufnahme können wie bei einem universellen Dateiverwaltungsprogramm (z. B. DBase) frei definiert und auf die jeweilige Nutzung angepaßt werden. Das Programm eignet sich zum Erstellen des gesamten Schriftverkehrs der bei der Auftragsabwicklung anfällt. Dazu gehören auch Werbeangebote und Rundschreiben an die einzelnen Kunden. Das kann durch Kombination mit C-Text geschehen, wobei die betreffenden Daten automatisch in einen Text eingesetzt werden.
Ein Produkt, das auch diesen Anwendungsbereich abdeckt ist das Programmpaket ST Manager Kunden, Lager, Rechnung das für 598,– DM erhältlich ist. Dieses Programmpaket wurde von der Firma SM geschrieben und wird wie der ST Manager Text von Atari vertrieben.
Ebenfalls über Atari wird ein Pascal-Compiler der Firma CCD angeboten. Er nennt sich 'GEM-DOS Pascal' und kostet 249,– DM. Es handelt sich hierbei um einen Ein-Pass Compiler, dessen Objekt-Module mit denen des C-Compilers und des Assemblers aus



dem Atari Entwicklungspaket kompatibel sind. Der Pascal Compiler erfüllt den ISO Standart, enthält aber noch einige Erweiterungen. Dazu gehören beispielsweise die Funktionen LOOP (Schleife ohne Abbruchbedingung), SEEK (Setzen eines Dateizeigers), CHAIN (Aufruf eines anderen Programmes) und Routinen zur Stringbehandlung auf die hier nicht näher eingegangen werden soll. Der Compiler verfügt über einen eigenen Editor und über Schnittstellen zum Einbeziehen von Assemblerprogrammen. Auch die internen Funktionen des GEM, des BIOS und des XBIOS können benutzt werden.
Ein Kalkulationsprogramm vom Kuma 'Spread' ist durch die Firma Softline zum Preis von 228,– DM zu beziehen. Es ist vergleichbar mit Multiplan, eines der bekanntesten Kalkulationsprogrammen für Personalcomputer. 'Spread' arbeitet mit der Maus und unterstützt die Fenstertechnik.
Mit dem amerikanischen 'VIP-Professional' wird der Softwaremarkt um ein hochwertiges Produkt bereichert. Es handelt sich hierbei um ein Programmpaket, das alle Eigenschaften des bekannten 'LOTUS 1-2-3' enthalten soll. Dazu gehört ein Dateiverwaltungs-, ein Kalkulations- und ein Textverarbeitungsprogramm. 'VIP-Professional' verfügt über die gleiche Bedienungsform wie 'LOTUS 1-2-3' und ist somit ideal für diejenigen die sich mit diesem Programm auskennen. Man kann allerdings die Kommandos auch über die gewohnten Drop-down Menüs anwählen. Dies ist einer der Verbesserungen die in diesem Programm vorgenommen worden sind. Dazu gehören auch die erweiterten grafischen Darstellungsmöglichkeiten, die dem Benutzer helfen einen Überblick über die Daten zu erhalten. Auf dem amerikanischen Softwaremarkt gibt es weitere Programme zur Textverarbeitung, Terminplanung und zur Finanzabwicklung.
Desweiteren existiert auch schon ein COBOL-Compiler für ca. 45 Dollar. Auch im Bereich der Utilities ist schon einiges verfügbar. Neben Diskettenmonitoren, Kopierprogrammen für Einzelfiles (lästiges Diskettenwechseln bei einem Laufwerk entfällt), existieren schon softwaremäßige Realisierungen von Druckerpuffern und einer RAM-Floppy. Diese RAM Floppy imitiert ein Laufwerk C, das seine Daten im freien Rechnerspeicher unterbringt. Dadurch erreicht man einen Speicherzugriff in kaum spürbarer Zeit. Die Kapazität dieses Pseudo-Laufwerkes ist frei einstellbar und nur von dem benötigten Programmspeicher abhängig.
Für die Datenfernübertragung stehen schon einige Programme zur Verfügung. Dazu gehört das 'ST-Comm' (R458,– DM, Softline) und das 'S-Term' (149,– DM, Ing. Büro Zoschke). Mit dem 'ST-Comm' kann man ein VT100 / VT102 Terminal simulieren, womit man Zugang zu Mailboxen und Datenbanken über Modem bekommt. Die Übertragungsgeschwindigkeit ist von 50 bis 19 200 Baud einstellbar. 'S-Term' bietet eine preislich günstige Möglichkeit der Kommunikation mit Mailboxen, Datenbanken oder anderen Rechnern. Eine ausführliche deutsche Anleitung wird mitgeliefert. Natürlich wird sich demnächst auch auf dem Spielesektor einiges tun. Momentan schon sind einige Programme vorhanden. Dazu gehören hauptsächlich Adventures (Ultima II, Crimson Crown, Transylvania), die allerdings meistens einen Farbmonitor benötigen und nicht unter 150,– DM erhältlich sind. Neben altbekannten Ballerspielen, wie Missle Command, Centipede, Battlezone oder Star Raiders, die auf dem amerikanischen Softwaremarkt angeboten werden (je 27 $), gibt es auch schon sinnvolle Denkspiele wie z. B. Reversi, zum allerdings recht hohen Preis von 160,– DM.

CP/M auf dem ATARI ST

Die größte Softwarequelle für den Atari ST öffnet sich durch den CP/M Emulator. Er verwandelt den Atari ST in eine CP/M 80 Maschine (Version 2.2), mit der der Zugriff auf weit verbreitete Software möglich ist. Unter CP/M gibt es eine nahezu unbeschränkte Vielfalt von guter Software. Dazu gehören die schon klassischen Programme: Wordstar, DBase und Turbo Pascal.
Die Aufgabe des Emulators besteht darin, die Routinen des CP/M-Betriebssystems, die in Z80 Maschinensprache geschrieben sind, dem TOS verständlich zu machen. Dazu muß der Z80 Maschinenkode in 68 000 Maschinenkode, den der Atari ST versteht, umgesetzt werden. Dies erledigt das Emulatorprogramm während des Programmablaufes. Man könnte den Z80 Emulator mit einem Interpreter vergleichen, der ebenfalls während des Programmablaufes aktiv wird. Ein Interpreter (z. B. Basic) greift bei Erkennen eines Befehls auf ein Maschinenunterprogramm zu, denn jedem Rechner ist primär nur Maschinensprache verständlich.
Dieser Vorgang ist immer mit einem Geschwindigkeitsverlust behaftet. Dies tritt auch im Falle des CP/M-Z80 Emulators auf. Aber hierbei kommt dem Atari ST zugute, daß der 68 000 Prozessor eine wesentlich höhere Arbeitsgeschwindigkeit hat und somit diesen Geschwindigkeitsverlust ausgleicht. Die Ausführungsgeschwindigkeit der Programme liegt in ähnlichen Bereichen wie bei reinen Z80-Computern.
Somit ist es der Firma Atari mit Hilfe des Emulators, der übrigens kopiert werden darf, gelungen ihrem Computer ein ausreichendes Softwareangebot zu erschließen, solange bis der Softwaremarkt mit speziell für den Atari ST geschriebener bzw. angepaßter Software nachgezogen hat. Das Problem ist das Diskettenformat, denn CP/M Software ist fast ausschließlich auf 5 1/4-Zoll Disketten verfügbar. Man kann davon ausgehen, daß in Zukunft viele Programme auf 3.5-Zoll Format erhältlich sind. Demjenigen der vielleicht schon CP/M Software besitzt hat die Möglichkeit ein großes Laufwerk (5.25) anzuschließen.
Die genaue Anleitung zum Anschluß eines solchen Laufwerkes finden Sie in dieser Ausgabe der ST-COMPUTER.

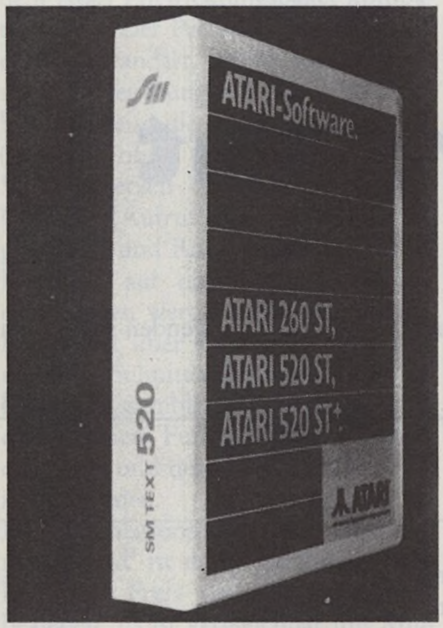
Software-Übersicht

Softwareübersicht

Die meisten Softwareprodukte sind über Fachhändler zu beziehen. Falls sie dort nicht erhältlich sind, wenden sie sich an folgende Adressen.

Hersteller/Vertrieb	Software		Preis
C-Soft GmbH	C-Text & C-Adress		250,–
Holzfällerstraße 1	C-Auftrag		500,–
8400 Regensburg			
FOCUS Computer			
GmbH	UCSD-Pascal Compiler	(p-System)	890,–
Friesenstraße 14	Fortran 77 Compiler	(p-System)	990,–
3000 Hannover 1	Basic Compiler	(p-System)	790,–
	p-System Tool Kit		490,–
	Modula-2 Compiler	(TDI)	890,–
	Pascal Compiler	(Metacomco)	475,–
	LISP Interpreter	(Metacomco)	795,–
	BCPL Compiler	(Metacomco)	445,–
	Lattice C-Compiler	(Metacomco)	595,–
	Makroassembler	(Metacomco)	395,–
	Pro Fortran	(Prospero)	900,–
SM-Soft	SM Text 520		239,–
Scherbaumstraße 33	SM Kunden, Lager, Rechnung		598,–
8000 München 83			
	(Vertrieb durch ATARI)		
VIP Technologies	VIP Professional		150 $
132 Aero Camino	VIP Analysis		k.A.
Santa Barabara	VIP Freelance (Text)		k.A.
CA 93117	VIP Consulat (Managment)		k.A.
G S T	Makro Assembler		40 Pfund
91 Highstreet	C-Compiler		79 Pfund
Longstanton			
Cambridge			
Softline (Vertrieb)	C-Compiler		348,–
Schwarzwaldstr. 8a	Makro-Assembler	KUMA	198,–
7602 Oberkirch	Spread	KUMA	228,–
	Ram Floppy	KUMA	198,–
	Textprogramm ST-Text		298,–
	ST-Comm		458,–
Ing. Büro Zoschke	S-Term		149,–
Postfach 1264			
8150 Holzkirchen			
	GEM-Dos Pascal		249,–
CCD			
Schöne Aussicht 41			
6229 Walluf	(Vertrieb durch Atari)		

ST Manager Text



ST Manager: Text (Version 1.1)

Was nützt der beste Computer ohne gute Software. Die meistgebrauchten Programme sind: Dateiverwaltung, Kalkulationsprogramme und Textverarbeitung. Ein Programm der letztgenannten Kategorie ist das SM-Text 520 von der deutschen Firma SM aus München. Das Programm wird von ATARI unter dem neuen Namen 'ST Manager Text' zum Preis von 239,– DM vertrieben.
Es handelt sich hierbei um ein hochwertiges Textprogramm, das nur wenige Wünsche unberücksichtigt läßt. Die Kommunikation mit dem Programm ist in deutscher Sprache gehalten. Beigelegt ist ein sehr ausführliches und gut gegliedertes deutsches Handbuch, das auch für den Computerlaien verständlich geschrieben ist, sodaß man schon nach kurzer Einarbeitungszeit den Umgang mit dem Textprogramm beherrscht.
Das Programm arbeitet nur zusammen mit dem monochromen Monitor.

Menüsteuerung

Sämtliche Textbearbeitungsfunktionen können aus verschiedenen Menüs aufgerufen werden. So muß man bei der Texteditierung keine schwerlich merkbare 'Control'-sequenzen (z. B. wie bei Wordstar) benutzen, sondern kann dem Rechner auf Menübasis seine Wünsche mitteilen. Das jeweilige Menü wird in der untersten Zeile dargestellt.

Übersichtlicher Bildschirm

Eine der angenehmsten Eigenschaften, die das Arbeiten mit dem ST-Textprogramm erleichtern, ist die übersichtliche Gestaltung des Bildschirms. In der ersten Zeile werden aktuelle Informationen dargestellt. Zum Beispiel wird einem neben der allgemein üblichen Koordinatenanzeige des Cursors angezeigt, ob man sich gerade im Transport-, Such-, Justier-, oder in einem anderen Modus befindet.
Besonders anschaulich ist, daß auf dem Bildschirm Fettschrift, invertierte, und unterstrichene Schrift dargestellt wird. Dadurch kann man die verschiedenen Schriftarten sofort auf dem Bildschirm kontrollieren, und nicht erst, wenn man einen Ausdruck vorliegen hat. Ebenso können diese Schrifttypen beliebig gemischt werden. Es wäre jedoch wünschenswert, wenn man auch Kursivschrift sowie Subscript und Superscript auf dem Bildschirm darstellen könnte.
Die Textbreite ist nicht auf 80 Zeichen pro Zeile beschränkt. Es können Texte bis zu einer Zeilenlänge von 254 Buchstaben bearbeitet werden, wobei der Textausschnitt bei mehr als 80 Zeichen zur Seite verschoben wird.

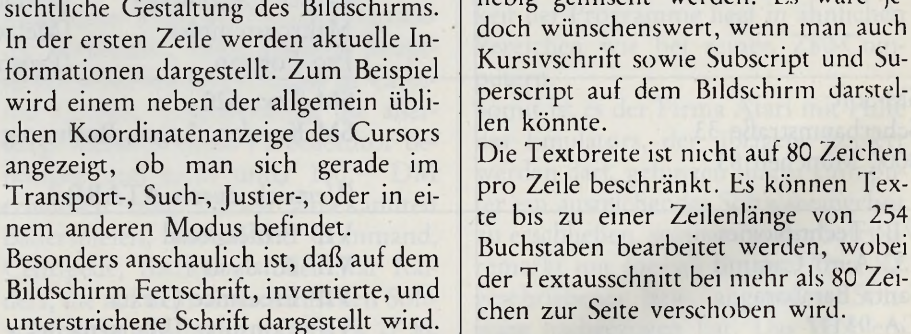


Bild 2: ausführliche Anleitung

Anleitung

Das mitglieferte Handbuch (siehe Bild 2), ist sehr ausführlich und außergewöhnlich übersichtlich aufgebaut. Die einzelnen Kapitel kann man, durch die eingefügten Ordnereinlagen sofort aufschlagen. Es enthält viele Beispiele und ist, auch für den unerfahrenen Anwender, verständlich geschrieben.

Umfangreiche Formatiermöglichkeit

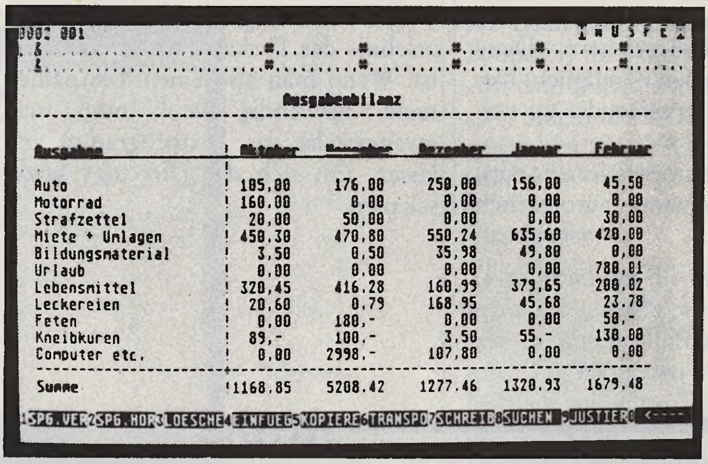
Zur Textformatierung stehen dem Benutzer folgende Möglichkeiten zur Verfügung:

- Texttabulatoren
- Zahlentabulatoren
- Spalteneinteilungen
- Zentrierung
- Randausgleich
- Texttrennung

Hiermit lassen sich einfach Tabellen oder strukturierte Texte erstellen (siehe Bild 3).
Mit der Spalteneinteilung ist es möglich, mehrreihige Texte zu erstellen, wie es zum Beispiel in Zeitschriften üblich ist. Des weiteren verfügt das Programm über eine Justierautomatik. Wenn ein Wort beim Justiervorgang (Blocken) über den rechten Rand hinausragt bietet das Textprogramm einen Trennvorschlag an, der auf einer einfachen, grammatikalischen Regel beruht. In der Regel wird vor einem Konsonanten mit folgendem Vokal der Trennstrich gesetzt. Man kann die Trennung natürlich auch an jeder anderen, selbst gewählten, Stelle vornehmen.

Prozeduren

Eine besondere Eigenschaft des ST-Textmanagers ist die Fähigkeit, Prozeduren zu verarbeiten. Unter einer Prozedur versteht man eine Folge von Befehlen. Dies können sowohl Bedienungsbefehle, als auch ganze Textabschnitte sein. Prozeduren können auf bestimmte Tasten gelegt werden und mit 'Control Return, beliebige Taste' aufgerufen werden. So können Redewendungen durch Tastendruck in den Text eingebaut werden. Wem das Anwählen oftbenutzter Menüfunktionen auf die Dauer zu mühselig ist, kann die



Ausgabenbilanz

Ausgaben	Oktober	November	Dezember	Januar	Februar
Auto	105,00	176,00	250,00	156,00	45,50
Motorrad	160,00	0,00	0,00	0,00	0,00
Strafzettel	20,00	50,00	0,00	0,00	30,00
Miete + Umlagen	450,30	470,80	550,24	635,60	420,00
Bildungsmaterial	3,50	0,50	35,98	49,80	0,00
Urlaub	0,00	0,00	0,00	0,00	780,01
Lebensmittel	320,45	416,28	160,99	379,65	200,02
Leckereien	20,60	0,79	168,95	45.68	23.78
Feten	0,00	180,-	0,00	0.00	50,-
Kneibkuren	89,-	100,-	3,50	55.-	130,00
Computer etc.	0,00	2998,-	107,80	0.00	0,00
Summe	1168.85	5208.42	1277.46	1320.93	1679.48

Bild 3: Tabellenbeispiel

gewünschte Befehlssequenz als Prozedur definieren und einer passenden Taste zuweisen. Als Beispiel sei das Laden und Speichern von Texten sowie Formatiervorgänge und das Drucken eines Dokumentes angesprochen. Aber Prozeduren können noch mehr. So kann das Erstellen eines Briefkopfes von einer Prozedur übernommen werden, welche nach dem Namen, der Anschrift und dem Datum fragt und sogleich die Daten im Briefkopf positioniert.
Eine Abspeicherung der Prozeduren ist möglich. Auf Wunsch können diese Prozeduren bei jedem Starten des



Bild 4: Lademenü

Textprogramms automatisch mitgeladen werden. Dies ermöglicht eine komfortable Tastaturdefinition, die sich jeder Benutzer auf seinen persönlichen Bedarf einstellen kann. Dadurch ist es dem Benutzer, der schon auf einem anderen Textverarbeitungsprogramm eingearbeitet ist (z. B. Wordstar), möglich deren spezielle Tastaturdefinitionen auch im ST-Textmanager zu übernehmen.

Diskettenoperationen

Beim Laden und Speichern von Texten bietet das Programm eine Vielfalt von Möglichkeiten. Es können nicht nur gesamte Texte geladen werden, sondern auch Teiltexte. Diese wiederum können in bestimmte Textstellen eingeschoben werden. Dies kann nicht nur zeilenweise, sondern ebenfalls spaltenweise, erfolgen (Bild 4).
ST-Text unterstützt alle externen Speicher, welcher aus einer oder zwei Floppies, einer Harddisk oder einer Ramfloppy bestehen kann. Das Spei-

chern von Texten (Bild 5) bietet auch die Möglichkeit, einen bestimmten Textausschnitt abzuspeichern. Diese Eigenschaften machen es möglich, Dokumente beliebig miteinander zu verknüpfen.
Bei allen Diskettenoperationen wird der aktuelle Textname automatisch eingetragen. Dies ist allerdings unbedingt notwendig, da das Inhaltsverzeichnis der Diskette nicht verfügbar ist. Wenn man also einen Textnamen nicht auswendig weiß, muß man zwangsweise das Textprogramm verlassen, um sich das Directory anzuschauen.



Bild 5: Speichermenü

Druckeranpassung

Bei der Anpassung von verschiedenen Druckern ist das Textprogramm allerdings wieder sehr komfortabel. In einem Druckeranpassungsprogramm kann man jeden einzelnen Buchstabencode, der zum Drucker geschickt wird, verändern (Bild 6). Die Steuersequenzen können ebenfalls auf den benutzten Drucker zugeschnitten werden. Darunter fallen die Steuerkodes für Fettschrift, Inversschrift, Unterstreichen, sowie die Sequenzen zur Druckerinitialisierung und des Linefeed (Bild 7). Dies ermöglicht den Anschluß jedes Druckers mit paralleler Schnittstelle an den Atari ST. Das Programm bietet dafür die Möglichkeit 10 verschiedene Druckereinstellungen abzuspeichern, wobei eine gekennzeichnete Einstellung automatisch eingeladen wird.
Auch das Druckmenü bietet einige Vielfalt von Einstellmöglichkeiten. Der Text kann stückweise oder spaltenweise ausgedruckt werden. Ebenso kann man beim Ausdruck den gesamten Text einrücken lassen, was bei gelochten Papier recht nützlich ist (Bild 8).

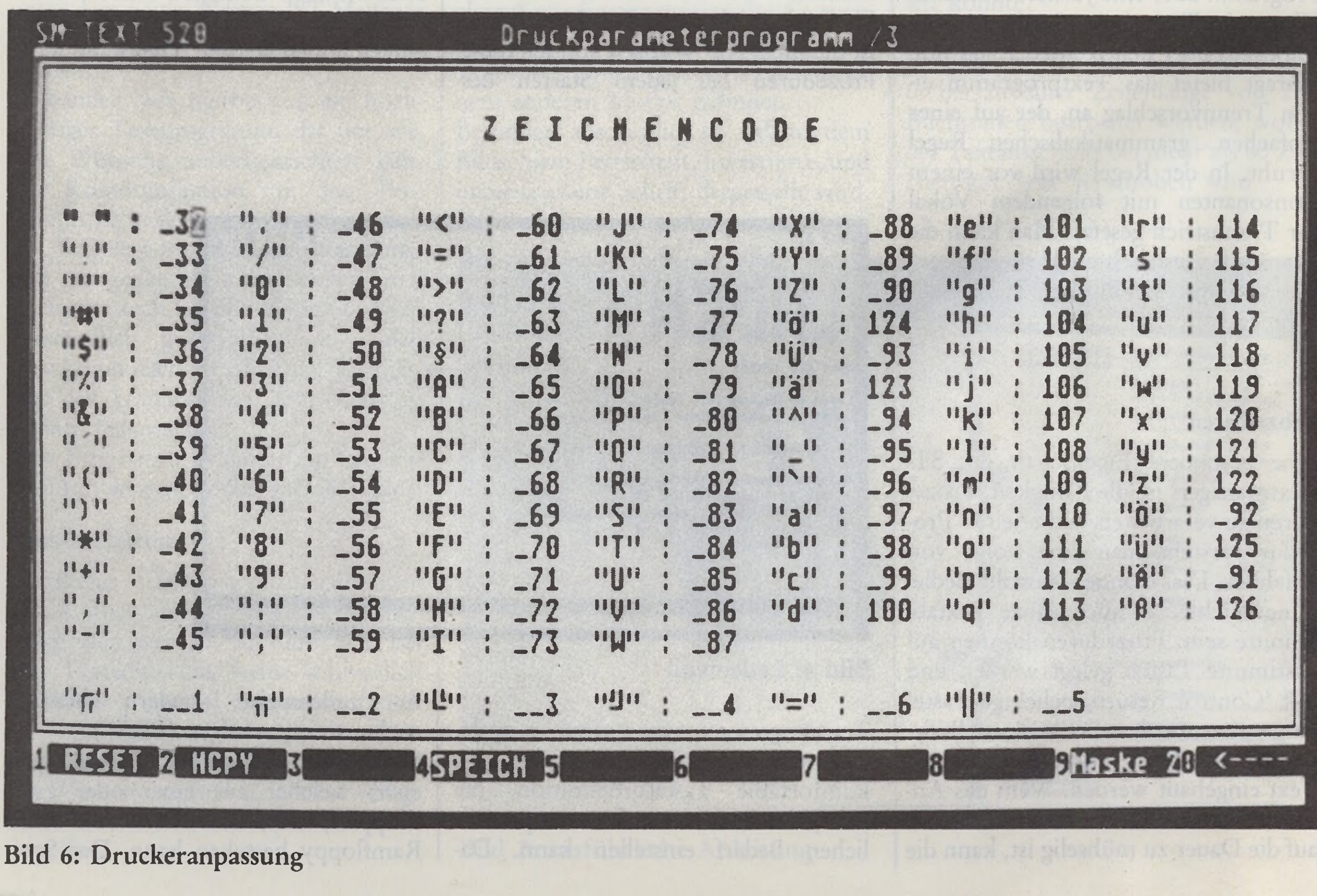


Bild 6: Druckeranpassung

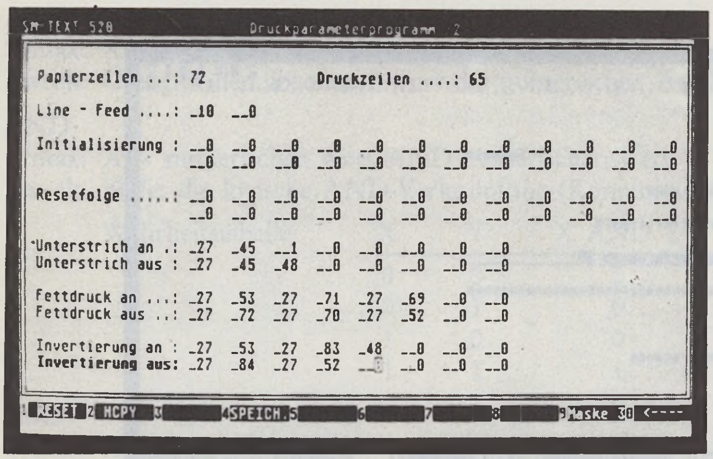


Bild 7: Druckeranpassung

Viele Einstellmöglichkeiten

Sämtliche Einstellungen, die der Benutzer tätigt, können abgespeichert werden. Dazu gehört die Druckeranpassung, die Druckereinstellung, die Tastenbelegungen durch die definierten Prozeduren und das Scrollverhalten. Die Scrolleinstellung gibt an, wann der Bildschirm in vertikaler Richtung scrollt, und ab welcher Spalte er sich bei Überschreitung des 80 Zeichen Bildschirmformates verschiebt. Der Benutzer kann sich so seinen Arbeitsplatz nach eigenen Wünschen gestalten und ist nicht auf vorgegebene Einstellungen angewisen.

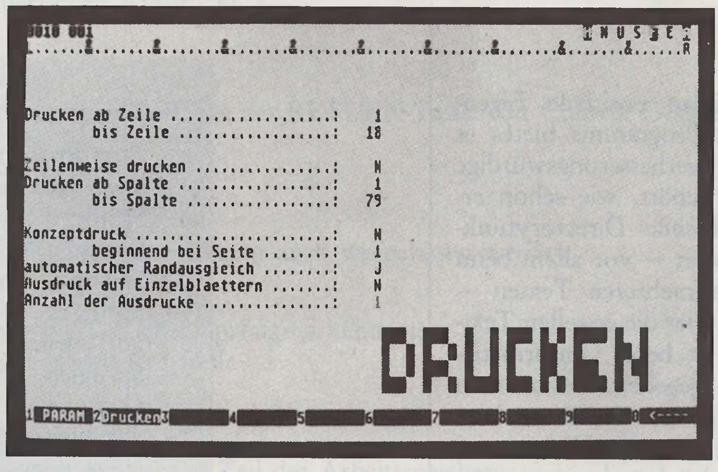


Bild 8: Druckermenü

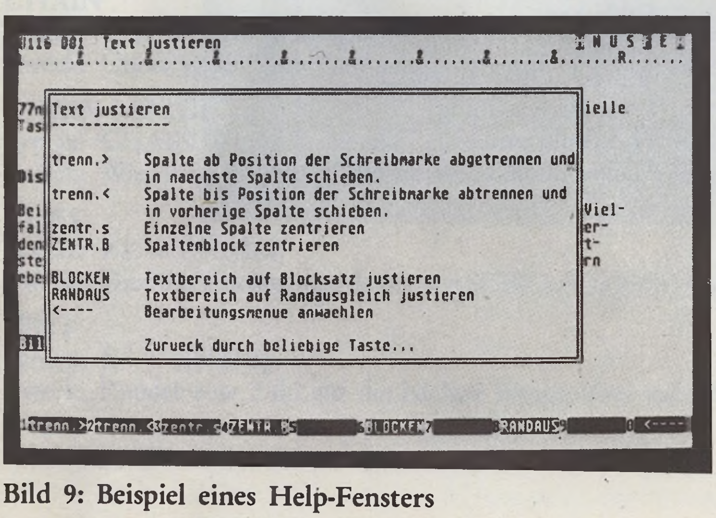


Bild 9: Beispiel eines Help-Fensters

Hilfestellungen

Bei Bedienungsunklarheiten kann der Benutzer die 'Help'-Taste benutzen. Auf dem Bildschirm erscheint ein Fenster, indem alle Punkte des derzeit angewählten Menüs kurz erklärt werden (Bild 9).

Textpuffer

Textabschnitte die kopiert, verschoben oder gelöscht werden sollen, werden in einem Pufferspeicher hinterlegt. Dies bietet den Vorteil, daß versehentlich gelöschte Textabschnitte wieder aus dem Puffer geholt werden können und somit nicht verloren gehen.

Rechenoperationen

Als Besonderheit unter den Textsystemen verfügt der ST Manager Text über einen Rechenmodus. Damit kann man umfangreiche mathematische Berechnungen durchführen. Dazu gehören neben den Grundrechenoperationen auch Speicherbefehle, Kehrwertbildung, Wurzelziehen, Quadratrechnen und Prozentrechnung.
Diese Funktionen kann man nebenbei als Taschenrechner benutzen, ebenso innerhalb der Texterstellung. Dadurch kann man Rechenvorgänge in den zu bearbeitenden Text einschieben.

Formularbearbeitung

Desweiteren bietet das Textprogramm die Möglichkeit, Daten in vorher erstellten Formularen einzusetzen. Dies ist zum Beispiel beim Erstellen von

Serienbriefen oder Rechnungen sehr nützlich (Bild 10).
Die Erstellung solcher Formulare wirft keine Probleme auf. Man markiert die Textfelder, in die nachher die betreffenden Daten eingegeben werden sollen, mit der betreffenden Anzahl von Füllzeichen ('★'). Der Cursor springt im Formularmodus automatisch in die definierten Textfelder und fügt die eingegebenen Daten in den Gesamttext ein. Während der Eingabe kann man den Cursor auch nicht aus diesen Textfenstern herausbewegen, wodurch Überschreibungen des Haupttextes verhindert werden. Diese Fähigkeit dürfte besonders für berufliche Anwender interessant sein, wobei auch die Rechenfähigkeit des Programmes hier erst richtig zur Geltung kommt.

Nachteile

Neben den vielen positiven Eigenschaften dieses Programms bietes es auch einige verbesserungswürdige Punkte. Dazu gehört, wie schon erwähnt, die fehlende Directoryfunktion. Sie erschwert – vor allem beim Bearbeiten von mehreren Texten – den Überblick über die erstellen Textfiles. Vorsicht ist beim Umformatieren (Blocken) gegeben. Sollte man beim Blocken anstatt ein überstehendes Wort zu trennen, fälschlicherweise die 'F10-Taste' drücken, kann man zusehen, wie der Rest des Textes oder Absatzes kommentarlos gelöscht wird. Dem unerfahrenen Anwender sei deshalb, wie bei allen neuen Programmen, empfohlen, seinen Text des öfteren zwischenzuspeichern, und natürlich das Handbuch gründlich zu studieren. Ebenso sollte man immer die richtige Diskette im Laufwerk haben, da das Programm die Eigenschaft hat, sich bei einer falschen Diskette zu verabschieden.
Bei der Darstellung von Steueranweisungen auf dem Bildschirm, die zum Beispiel die Zeichensätze auf dem Drucker umschalten, erscheinen nur schwer, bzw. überhaupt nicht, unterscheidbare Zeichen. Dazu gehören neben den Druckeranweisungen auch die Umrahmungszeichen, die vor allem bei Tabellen ihre Anwendung finden.

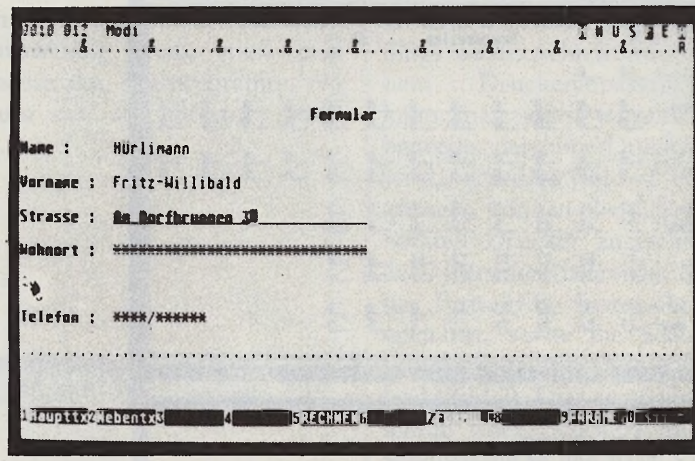


Bild 10: Beispiel der Formularbearbeitung

Befehlsübersicht:

- Text löschen/einfügen
- Text verschieben/kopieren
- Springen Seite vor/zurück
- Springen Textanfang/ende
- Springen Absatz vor/zurück
- Springen horizontal Wort vor/zurück
- Springen horizontal Tabulator vor/zurück
- Springen horizontal Zeilenanfang/ende
- Suchen und Ersetzen von Wörtern
- Text justieren: Trennen, Blocken, Randausgleich (rechtsbündig)
- Abspeichern, Laden, Zusammenfügen von Texten (auch Textteile)
- Tabulatoren: Text- und Zahlentabulator, Randverschiebung
- Umfangreiche Druckereinstellmöglichkeiten
- Rechenfunktionen (+, –, ★, /, %, Wurzel, Quadrat, etc...)
- Formularerstellung
- Hilfsfunktion
- Haupt- und Nebentextverarbeitung

BASIC-BEFEHLSSYNTAX

ABS
Syntax: X = ABS(Y)
Zweck: Übergibt den absoluten Wert des numerischen Ausdrucks Y.

AND
Syntax: X = numerischer Wert AND numerischer Wert
Zweck: Bildet die logische AND-Verknüpfung (Konjunktion) zweier numerischer Werte.

Wahrheitstabelle:

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

ASC
Syntax: X = ASC(Y$)
Zweck: Übergibt den ASCII-Code für das erste Zeichen der Zeichenkette Y$.

ATN
Syntax: X = ATN(Y)
Zweck: Übergibt den Arcustangens des numerischen Ausdrucks Y.

AUTO
Syntax: AUTO Startzeilennummer, Schrittweite
Zweck: Automatische Zeilennummerierung bei jedem Betätigen der RETURN-Taste. Mit Control G kann man AUTO abschalten.

BLOAD
Syntax: BLOAD "Name.Extension",Adresse
Zweck: Lädt eine Datei in den Arbeitsspeicher. Die Adresse kann auch weggelassen werden

BREAK
Syntax: BREAK Zeilennummer oder BREAK symbolischer Sprungmarkennamen
Zweck: Unterbricht den Programmablauf an angegebener Stelle

BSAVE
Syntax: BSAVE "Name.Extension",Adresse,Länge
Zweck: Speichert einen durch Adresse und Länge definierten Teil des Arbeitsspeichers.

CALL
Syntax: CALL numerische Variable (Parametervariable,Parametervariable,...)
Zweck: Ruft ein Maschinensprachenunterprogramm auf.

CDBL
Syntax: X = CDBL(Y)
Zweck: Wandelt den numerischen Ausdruck Y in eine Zahl doppelter Genauigkeit um.

CHAIN
Syntax: CHAIN "Name.Extension",Startzeile im aufgerufenen Programm,ALL
Zweck: Überträgt die Kontrolle an ein aufgerufenes Programm und übergibt alle Variablen aus dem laufenden Programm.

CHAIN MERGE
Syntax: CHAIN MERGE "Name.Extension",Startzeile,DELETE Startzeile-Endzeile
Zweck: Wie oben, nur werden zuerst die angegebenen Zeilen im alten Programm gelöscht.

CHR$
Syntax: X$ = CHR$(n)
Zweck: Gibt das Zeichen aus, das dem ASCII-Code der eingegebenen Zahl n entspricht (n muß zwischen 0 und 255 liegen).

CINT
Syntax: X = CINT(Y)
Zweck: Rundet eine Zahl auf die nächste Integer-Zahl auf oder ab.

CIRCLE
Syntax: CIRCLE X,Y,Radius oder CIRCLE X,Y,Radius,Anfangswinkel,Endwinkel
Zweck: Zeichnet einen Kreis oder Kreisausschnitt (Anfangs- und Endwinkel sind Winkel in Grad ★ 10).

CLEAR
Syntax: CLEAR oder CLEAR,Anzahl des noch verfügbaren Speicherplatz in Bytes
Zweck: Löschen aller numerischen Variablen und Zeichenkettenvariablen oder definieren der Größe des Arbeitsspeichers des BASIC.

CLEARW
Syntax: CLEARW 0 oder CLEARW 1 oder CLEARW 2 oder CLEARW 3
Zweck: Löscht den Inhalt eines Fensters.
EDIT-Fenster = 0 LIST-Fenster = 1 OUTPUT-Fenster = 2 COMMAND-Fenster = 3

CLOSE
Syntax: CLOSE Dateinummer oder CLOSE #Dateinummer
Zweck: Schließt Ein- oder Ausgaben oder Dateien.

CLOSEW
Syntax: CLOSEW 0 oder CLOSEW 1 oder CLOSEW 2 oder CLOSEW 3
Zweck: Schließt ein Fenster.
EDIT-Fenster = 0 LIST-Fenster = 1 OUTPUT-Fenster = 2 COMMAND-Fenster = 3

COLOR
Syntax: COLOR Textfarbe,Füllfarbe,Linienfarbe,Index,Stil
Zweck: Stellt die Text-, Füll- und Zeichenfarbe und die Füllmuster ein.

COMMON
Syntax: COMMON Variable,Variable,...
Zweck: Übergibt Variablen an ein mit CHAIN aufgerufenes Programm.

CONT
Syntax: CONT
Zweck: Nimmt einen mit BREAK, STOP oder CONTROL G unterbrochenen Programmlauf wieder auf.

COS
Syntax: X = COS(Y)
Zweck: Gibt den Cosinuswert des numerischen Ausdrucks Y aus.

CSNG
Syntax: X = CSNG(Y)
Zweck: Wandelt den numerischen Ausdruck Y in eine Zahl einfacher Genauigkeit um.

CVI,CVS,CVD
Syntax: X = CVI (2-Byte-Zeichenkette)
X = CVS (4-Byte-Zeichenkette)
X = CVD(8-Byte-Zeichenkette)
Zweck: Wandelt den Inhalt von Zeichenkettenvariablen in numerische Werte um.

DATA
Syntax: DATA Konstante,Konstante,...
Zweck: Definiert eine Liste von Konstanten, die mit READ gelesen werden können.

DEF FN
Syntax: DEF FN Funktionsname(Argument,Argument,...)=Definition
Zweck: Definiert eine Funktion und gibt ihr einen Namen.

DEFDBL,DEFINT,DEFSNG
Syntax: DEFTyp Buchstabe oder DEFTyp Buchstabe-Buchstabe
Zweck: Gibt einen bzw. einen Bereich von Buchstaben, der als Realzahl definiert werden soll.
DEFDBL definiert eine Realzahl doppelter Genauigkeit.
DEFINT definiert eine Integerzahl.
DEFSNG definiert eine Realzahl einfacher Genauigkeit.

DEFSTR
Syntax: DEFTyp Buchstabe oder DEFTyp Buchstabe-Buchstabe
Zweck: Gibt einen bzw. einen Bereich von Buchstaben, der als Zeichenkette definiert werden soll.

DELETE
Syntax: DELETE Zeilennummer-Zeilennummer
Zweck: Löscht den angegebenen Zeilenbereich aus dem Programm.

DIM
Syntax: DIM Variable(Index) oder DIM Variable(Index,Index)
Zweck: Reserviert ein durch Index in der Größe begrenztes Feld im Arbeitsspeicher.

DIR
Syntax: DIR oder DIR Laufwerk: oder DIR Laufwerk:Dateiname.Extension
Zweck: Auflisten von einer oder mehrerer Dateien der Diskette. Es können auch die Joker ★ und ? eingesetzt werden.

EDIT
Syntax: EDIT oder ED oder EDIT Zeilennummer oder ED Zeilennummer
Zweck: Ruft den BASIC-Editor auf.

ELLIPSE
Syntax: ELLIPSE hor.Mittelpunkt,vert.Mittelpunkt,hor.Radius,vert.Radius
Zweck: Zeichnet eine Ellipse bzw. falls man noch einen Anfangs- und Endwinkel anhängt einen Ellipsenausschnitt.

END
Syntax: END
Zweck: Beendet die Programmausführung, schließt alle Dateien und kehrt zur Befehlsebene zurück.

EOF
Syntax: EOF (Dateinummer)
Zweck: Zeigt die Bedingung Dateiende (End of file) an.

EQV
Syntax: X = numerischer Wert EQV numerischer Wert
Zweck: Bildet die logische EQV-Verknüpfung (Äquivalenz) zweier numerischer Werte.

Wahrheitstabelle:

X	Y	X EQV Y
0	0	1
0	1	0
1	0	0
1	1	1

ERA
Syntax: ERA Dateiname.Extension oder ERA Laufwerk:Dateiname.Extension
Zweck: Löscht eine Datei von Diskette.

ERASE
Syntax: ERASE Variable
Zweck: Löscht Feldvariablen.

ERL
Syntax: X = ERL
Zweck: Die Variable ERL enthält die Zeilennummer der Zeile, in der der letzte Fehler aufgetreten ist.

ERR
Syntax: X = ERR
Zweck: Die Variable ERR enthält den Fehlercode für den letzten Fehler.

ERROR
Syntax: ERROR X
Zweck: Mit ERROR kann man den Auftritt eines BASIC-Fehlers simulieren oder einen eigenen Fehlercode definieren.

EXP
Syntax: X = EXP(Y)
Zweck: Ergibt die Konstante e, erhöht um den Exponenten Y.

FIELD
Syntax: FIELD #Dateinummer,Länge AS Zeichenkettenvariable,Länge AS ...
Zweck: Legt Platz für Variablen in einem Puffer für Direktzugriffs-Dateien an.

FILL
Syntax: FILL X-Koordinate,Y-Koordinate
Zweck: Füllt von Punkt X,Y aus eine Fläche mit einer Farbe aus, die vorher mit dem COLOR-Befehl festgelegt wurde.

FIX
Syntax: X = FIX(Y)
Zweck: Verkürzt eine Realzahl auf eine Integerzahl.

FLOAT
Syntax: X = FLOAT(Y)
Zweck: Wandelt eine Integerzahl in eine Realzahl mit einfacher Genauigkeit um.

FOLLOW
Syntax: FOLLOW Variable,Variable,...
Zweck: Verfolgt die Werte von Programmvariablen und gibt sie auf den Bildschirm aus.

FOR
Syntax: FOR Variable=X TO Y oder FOR Variable=X TO Y STEP Z
Zweck: Führt eine Folge von Befehlen in einer Schleife aus, wobei die Zahl der Ausführungen angegeben ist.

FRE
Syntax: X = FRE(0)
Zweck: Übergibt die Anzahl der nicht verwendeten Bytes des Arbeitsspeichers.

FULLW
Syntax: FULLW 0 oder FULLW 1 oder FULLW 2 oder FULLW 3
Zweck: öffnet ein Fenster auf volle Bildschirmgröße.
EDIT-Fenster = 0 LIST-Fenster = 1 OUTPUT-Fenster = 2 COMMAND-Fenster = 3

GEMSYS
Syntax: GEMSYS(AES Op Code)
Zweck: Erlaubt mittels des PEEK-Befehls einen Zugriff auf die AES-Kontrollfelder.

GET#
Syntax: GET #Dateinummer oder GET #Dateinummer,Datensatznummer
Zweck: Liest einen Datensatz aus einer Direktzugriffs-Datei in einen Puffer.

GOSUB
Syntax: GOSUB Zeilennummer oder GOSUB Sprungmarke
Zweck: Verzweigt in eine Unterprogrammroutine.

GOTO
Syntax: GOTO Zeilennummer oder GOTO Sprungmarke
Zweck: Verzweigt unbedingt nach der angegebenen Zeilennummer bzw. Sprungmarke.

GOTOXY
Syntax: GOTOXY Spalte,Reihe
Zweck: Positionieren des Cursors in einer bestimmten Spalte und Reihe.

HEX$
Syntax: X$= HEX$(Dezimalwert)
Zweck: Übergibt den Hexadezimalwert einer Dezimalzahl.

IF ... THEN ...
Syntax: IF logischer Ausdruck THEN Anweisung oder
IF logischer Ausdruck THEN Anweisung ELSE Anweisung
Zweck: Ändert den Programmablauf, abhängig vom Ergebnis des logischen Ausdrucks.

IMP
Syntax: X = numerischer Wert IMP numerischer Wert
Zweck: Bildet die logische IMP-Verknüpfung (Implikation) zweier numerischer Werte.

Wahrheitstabelle:

X	Y	X IMP Y
0	0	1
0	1	1
1	0	0
1	1	1

INP
Syntax: X = INP(Datenkanal-Nummer)
Zweck: Gibt einen Byte-Wert von einem bestimmten Datenkanal aus.

INPUT
Syntax: INPUT Variable oder INPUT "Text";Variable
Zweck: Ermöglicht Dateneingabe während des Programmablaufs und ordnet diese Daten einer oder mehreren Variablen zu.

INPUT#
Syntax: INPUT#Dateinummer,Variable
Zweck: Liest Daten aus einer sequentiellen Einheit oder Datei und weist sie Programmvariablen zu.

INPUT$
Syntax: X$ = INPUT$(Anzahl der Zeichen) oder
X$ = INPUT$(Anzahl der Zeichen),#Dateinummer
Zweck: Übergibt eine Anzahl von Zeichen, die von der Tastatur oder aus einer Datei gelesen wurden.

INSTR
Syntax: X = INSTR(Zeichenkette,Suchzeichenkette) oder
X = INSTR(Startposition,Zeichenkette,Suchzeichenkette)
Zweck: Sucht eine Suchzeichenkette in einer anderen Zeichenkette und übergibt deren Position an eine Variable. Es ist auch möglich, erst ab einer bestimmten Startposition mit der Suche zu beginnen.

INT
Syntax: X = INT(numerischer Ausdruck)
Zweck: Wandelt einen numerischen Ausdruck in eine Integer-Zahl um.

KILL
Syntax: KILL Name.Extension
Zweck: Löscht eine Datei von Diskette.

LEFT$
Syntax: X$ = LEFT$(Zeichenkette,Anzahl der Zeichen)
Zweck: Übergibt die von links gerechnet durch die Anzahl bestimmten Zeichen einer Zeichenkette an X$.

LEN
Syntax: X = LEN(Zeichenkette)
Zweck: Übergibt die Länge einer Zeichenkette.

LET
Syntax: LET Variable=Ausdruck
Zweck: Ordnet den Wert eines Ausdrucks einer Variablen zu.

LINE INPUT
Syntax: LINE INPUT;Variable oder LINE INPUT "Text";Variable
Zweck: Wie INPUT. Es ist aber möglich eine ganze Zeile (255 Zeichen) als Antwort zu geben. Trennzeichen werden dabei ignoriert.

LINE INPUT#
Syntax: LINE INPUT#;Variable oder LINE INPUT# "Text";Variable
Zweck: Wie INPUT#. Es ist aber möglich eine ganze Zeile (255 Zeichen) als Antwort zu geben. Trennzeichen werden dabei ignoriert.

LINEF
Syntax: LINEF X1,Y1,X2,Y2
Zweck: Zeichnet eine Linie mit den Anfangskoordinaten X1,Y1 und den EndkoordinatenX2,Y2.

LIST
Syntax: LIST oder LIST Zeilennummer-Zeilennummer
Zweck: Listet das im Speicher befindliche BASIC-Programm ganz oder teilweise im LIST-Fenster.

LLIST
Syntax: LLIST oder LLIST Zeilennummer-Zeilennummer
Zweck: Wie LIST, nur wird das Programm auf einem Drucker ausgegeben.

LOAD
Syntax: LOAD "Dateiname"
Zweck: Lädt eine Datei von Diskette in den Arbeitsspeicher.

LOC
Syntax: X = LOC(Dateinummer)
Zweck: Gibt entweder eine Datensatznummer oder die Anzahl der Bytes, die von einer Datei gelesen bzw. in eine Datei geschrieben wurden, aus.

LOF
Syntax: X = LOF(Dateinummer)
Zweck: Übergibt die Anzahl der Bytes, die einer Datei zugeordnet sind.

LOG
Syntax: X = LOG(numerischer Ausdruck)
Zweck: Gibt den natürlichen Logarithmus einer Zahl aus.

LOG10
Syntax: X = LOG10(numerischer Ausdruck)
Zweck: Gibt den Zehnerlogarithmus einer Zahl aus.

LPOS
Syntax: X = LPOS(numerischer Ausdruck)
Zweck: Übergibt die laufende Position des Druckkopfes im Druckerpuffer für einen arbeitenden Drucker aus.

LPRINT
Syntax: LPRINT Variable oder LPRINT "Text";Variable
Zweck: Druckt Daten auf einem Drucker aus.

LPRINT USING
Syntax: LPRINT USING Formatzeichenkette;Variable
Zweck: Druckt Daten formatiert auf einem Drucker aus.

LSET
Syntax: LSET Zeichenkettevariable=Zeichenkettevariable
Zweck: Übergibt Daten in einen Dateipuffer für Direktzugriff. Die Daten werden linksbündig gespeichert und auf die mit FIELD definierte Datenlänge mit Leerstellen aufgefüllt.

MERGE
Syntax: MERGE Name.Extension
Zweck: Mischt die Zeilen einer BASIC-Datei mit dem im Arbeitsspeicher befindlichen Programm.

MID$
Syntax: X$ = MID$(Zeichenkette,Anfangspunkt,Länge)
Zweck: Gibt ein durch Anfangspunkt und Länge bestimmtes Stück einer Zeichenkette aus. Es ist allerdings auch möglich, einer Zeichenkette eine andere durch MID$ definierte Zeichenkette zuzuordnen.

MKD$,MKI$,MKS$
Syntax: X$ = MKD$(Ausdruck mit doppelter Genauigkeit)
X$ = MKI$ (ganzzahliger Ausdruck)
X$ = MKS$ (Ausdruck mit einfacher Genauigkeit)
Zweck: Wandelt ASCII-Zeichenketten, die Zahlen darstellen, in Byte-Zeichenketten um.

MOD
Syntax: Numerischer Wert MOD numerischer Wert
Zweck: Übergibt den Rest einer Integerdivision.

NAME
Syntax: NAME "alter Name.Extension" AS "neuer Name.Extension"
Zweck: Umbenennen einer Datei auf Diskette.

NEW
Syntax: NEW Name.Extension
Zweck: Löscht eine Datei aus dem Arbeitsspeicher.

NEXT
Syntax: NEXT numerische Variable
Zweck: Gibt das Ende einer FOR ... NEXT Schleife an.

NOT
Syntax: Numerischer Wert NOT numerischer Wert
Zweck: Bildet die logische NOT-Verknüpfung (Komplement) zweier numerischer Werte.

Wahrheitstabelle:

X	NOT X
0	1
1	0

OCT$
Syntax: X$ = OCT$(Dezimalwert)
Zweck: Übergibt den Oktalwert einer Dezimalzahl.

OLD
Syntax: OLD Dateiname
Zweck: Lädt eine bestehende Programmdatei in den Arbeitsspeicher.

ON ERROR GOTO
Syntax: ON ERROR GOTO Zeilennummer
Zweck: Übergibt die Programmkontrolle bei Auftreten eines Fehlers an ein Unterprogramm, dessen Anfang durch die Zeilennummer bestimmt wird.

ON ... GOSUB
Syntax: ON numerische Variable GOSUB Zeilennummer,Zeilennummer,...
Zweck: Verzweigt zu einer von mehreren Zeilennummern, die den Anfang von Unterprogrammroutinen angeben, in Abhängigkeit vom Wert der numerischen Variablen.

ON ... GOTO
Syntax: ON numerische Variable GOTO Zeilennummer,Zeilennummer,...
Zweck: Verzweigt zu einer von mehreren Zeilennummern in Abhängigkeit vom Wert der numerischen Variablen.

OPEN
Syntax: OPEN "Modus",#Dateinummer,Dateiname.Extension oder
OPEN "R",#Dateinummer,Dateiname.Extension,Datensatzlänge
Zweck: Ermöglicht die Ein- und Ausgabe an und von einer Datei oder Einheit. Die verschiedenen Modi haben folgende Bedeutung:
"I" öffnet eine sequentielle Datei zum Lesen.
"O" öffnet eine sequentielle Datei zum Schreiben.
"R" öffnet eine Direktzugriffs-Datei zum Lesen und Schreiben.

OPENW
Syntax: OPENW 0 oder OPENW 1 oder OPENW 2 oder OPENW 3
Zweck: öffnet ein Fenster auf eine vorgewählte Größe.
EDIT-Fenster = 0 LIST-Fenster = 1 OUTPUT-Fenster = 2 COMMAND-Fenster = 34

OPTION BASE
Syntax: OPTION BASE 0 oder OPTION BASE 1
Zweck: Definiert den kleinsten Wert für indizierte Variablen.

OR
Syntax: Numerischer Wert OR numerischer Wert
Zweck: Bildet die logische OR-Verknüpfung (Disjunktion) zweier numerischer Werte.

Wahrheitstabelle:

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

OUT
Syntax: OUT Ausgabeadresse,Integerausdruck für zu sendende Daten
Zweck: Sendet ein Byte an einen Datenausgabekanal.

PCIRCLE
Syntax: PCIRCLE X,Y,Radius oder PCIRCLE X,Y,Radius,Anfangswinkel,Endwinkel
Zweck: Zeichnet einen ausgefüllten Kreis oder Kreisausschnitt (Anfangs- und Endwinkel sind Winkel in Grad ★ 10).

PEEK
Syntax: X = PEEK(Speicheradresse)
Zweck: Liest ein Wort aus dem Speicher aus.

PELLIPSE
Syntax: PELLIPSE hor.Mittelpunkt,vert.Mittelpunkt,hor.Radius,vert.Radius
Zweck: Zeichnet eine ausgefüllte Ellipse bzw. falls man noch einen Anfangs- und Endwinkel anhängt einen ausgefüllten Ellipsenausschnitt.

POKE
Syntax: POKE Speicheradresse,Byte
Zweck: Schreibt ein Wort in den Speicher.

POS
Syntax: X = POS(0)
Zweck: Übergibt die derzeitige Spaltenposition des Cursors.

PRINT
Syntax: PRINT Variable oder PRINT "Text";Variable
Zweck: Gibt Daten im OUTPUT-Fenster aus.

PRINT USING
Syntax: PRINT USING Formatzeichenkette,Variable
Zweck: Gibt Daten formatiert im OUTPUT-Fenster aus.

PRINT#
Syntax: PRINT#Dateinummer,Variable oder PRINT#Dateinummer,"Text";Variable
Zweck: Gibt Daten sequentiell an eine Datei aus.

PRINT#...,USING
Syntax: PRINT#Dateinummer,USING Formatzeichenkette,Variable
Zweck: Gibt Daten formatiert sequentiell an eine Datei aus.

PUT#
Syntax: PUT#Dateinummer,Datensatznummer
Zweck: Schreibt einen Datensatz aus dem Puffer für Direktzugriffsdateien auf Diskette.

QUIT
Syntax: QUIT
Zweck: Beendet das BASIC und kehrt zum GEM-Desktop zurück.

RANDOMIZE
Syntax: RANDOMIZE numerischer Ausdruck
Zweck: Übergibt einen Anfangswert für den Zufallszahlengenerator.

READ
Syntax: READ Variable oder READ Variable,Variable
Zweck: Liest die Wert der DATA-Anweisung und ordnet sie der bzw. den Variablen zu.

REM oder '
Syntax: REM Bemerkung oder 'Bemerkung
Zweck: Erlaubt das Einfügen von Bemerkungen in das Programm, ohne daß dadurch der Programmablauf gestört wird.

RENUM
Syntax: RENUM oder RENUM neue erste Zeilennummer,,Schrittweite oder
RENUM neue erste Zeilennummer,Startzeilennummer,Schrittweite
Zweck: Nummeriert das Programm ganz oder teilweise neu durch. Wenn keine Parameter angegeben werden, wird die erste neue Zeilennummer und die Schrittweite mit 10 angenommen.

REPLACE
Syntax: REPLACE Name.Extension oder REPLACE Name.Extension,Zeilennummer-Zeilennummer
Zweck: Ersetzt eine alte Version einer Datei voll oder teilweise durch eine neue Version.

RESET
Syntax: RESET
Zweck: Schließt alle Dateien ab und löscht den Systempuffer.

RESTORE
Syntax: RESTORE oder RESTORE Zeilennummer
Zweck: Setzt den Zeiger für DATA-Anweisungen zu lesen ganz oder auf eine bestimmte Zeilennummer zurück.

RESUME
Syntax: RESUME Zeilennummer oder RESUME (0)
Zweck: Fährt nach einem aufgetretenen Fehler entweder an der angegebenen Stelle fort oder fährt mit der Anweisung fort die den Fehler verursachte.

RESUME NEXT
Syntax: RESUME NEXT
Zweck: Fährt nach einem aufgetretenen Fehler mit der Anweisung fort, die hinter der Anweisung steht, die den Fehler verursachte.

RETURN
Syntax: RETURN
Zweck: Kehrt von einer mit GOSUB aufgerufenen Unterroutine zurück und macht mit der nächsten Anweisung weiter.

RIGHT$
Syntax: X$ = RIGHT$(Zeichenkette,Anzahl der Zeichen)
Zweck: Übergibt die von rechts gerechnet durch die Anzahl bestimmten Zeichen einer Zeichenkette an X$.

RND
Syntax: X = RND oder X = RND(0) doer X = RND(Zahl oder numerische Variable)
Zweck: Erzeugt eine Zufallszahl, deren Anfangswert zuvor mit RANDOMIZE bestimmt wurde. Der Anfangswert kann aber auch mit einer negativen Zahl bzw. numerischen Variablen definiert werden. RND(0) wiederholt die zuletzt erzeugte Zahl. Ist die Zahl bzw. numerische Variable positiv oder es wird nichts hinter RND angegeben, wird die nächste Zufallszahl in Folge erzeugt.

RSET
Syntax: RSET Zeichenkettevariable=Zeichenkettevariable
Zweck: Übergibt Daten in einen Dateipuffer für Direktzugriff. Die Daten werden rechtsbündig gespeichert und auf die mit FIELD definierte Datenlänge mit Leerstellen aufgefüllt.

RUN
Syntax: RUN oder RUN,Zeilennummer oder RUN Name.Extension oder
RUN Name.Extension,Zeilennummer
Zweck: Beginnt die Programmausführung. Wenn ein Dateiname angegeben wird, wird das Programm geladen und automatisch gestartet. Beide Möglichkeiten können auch erst ab einer bestimmten Zeilennummer gestartet werden.

SAVE
Syntax: SAVE Dateiname oder SAVE Dateiname,Zeilennummer-Zeilennummer
Zweck: Speichert das Programm ganz oder teilweise auf Diskette.

SGN
Syntax: X = SNG(numerischer Ausdruck)
Zweck: Übergibt das Vorzeichen des numerischen Ausdrucks.
Ist der numerische Ausdruck positiv, wird eine 1 übergeben.
Ist der numerische Ausdruck gleich 0, wird eine 0 übergeben.
Ist der numerische Ausdruck negativ, wird eine −1 übergeben.

SIN
Syntax: X = SIN(numerischer Ausdruck)
Zweck: Errechnet den Sinuswert für einen Wert im Bogenmaß.

SOUND
Syntax: SOUND Stimme,Lautstärke,Note,Oktave,Tondauer
Zweck: Erzeugt einen Ton.

SPACE$
Syntax: X$ = SPACE$(numerischer Ausdruck)
Zweck: Gibt eine durch den numerischen Ausdruck definierte Anzahl von Leerzeichenketten aus.

SPC
Syntax: PRINT SPC(numerischer Ausdruck)
Zweck: Gibt eine durch den numerischen Ausdruck definierte Anzahl von Leerstellen in einer PRINT-Anweisung aus.

SQR
Syntax: X = SQR(numerischer Ausdruck)
Zweck: Übergibt die Quadratwurzel eines numerischen Ausdrucks.

STEP
Syntax: STEP oder STEP,Zeilennummer oder STEP Name.Extension oder STEP Name.Extension,Zeilennummer
Zweck: Führt ein Programm zeilenweise aus. Falls ein Dateiname angegeben wurde, wird zuerst das Programm geladen und dann ausgeführt. Es ist auch möglich ein Programm erst ab einer bestimmten Zeilennummer zu starten.

STOP
Syntax: STOP
Zweck: Bricht den Programmablauf ab und kehrt in das COMMAND-Fenster zurück.

STR$
Syntax: X$ = STR$(numerischer Ausdruck)
Zweck: Übergibt die Zeichenkettendarstellung eines numerischen Ausdrucks.

STRING$
Syntax: X$ = STRING$(numerischer Ausdruck,Zeichenkette) oder
X$ = STRING$(numerischer Ausdruck,numerischer Ausdruck)
Zweck: Übergibt eine Zeichenkette, deren Länge durch den ersten numerischen Ausdruck definiert ist, und deren Zeichen alle aus dem ASCII-Code des zweiten numerischen Ausdrucks oder der Zeichenkette bestehen.

SWAP
Syntax: SWAP Variable,Variable
Zweck: Vertauscht die Werte zweier Variablen.

SYSTAB
Syntax: X = PEEK(SYSTAB+OFFSET)
Zweck: Liest den Zeiger, der auf die Systemtabelle zeigt.

SYSTEM
Syntax: SYSTEM
Zweck: Verläßt das BASIC und kehrt zum GEM-Desktop zurück.

TAB
Syntax: PRINT TAB(numerischer Ausdruck)
Zweck: Setzt einen Tabulator an der durch den numerischen Ausdruck angegebenen Stelle.

TAN
Syntax: X = TAN(numerischer Ausdruck)
Zweck: Gibt den Tangens eines numerischen Ausdrucks aus.

TRACE
Syntax: TRACE oder TRACE Zeilennummer oder TRACE Zeilennummer-Zeilennummer
Zweck: Zeigt während des Programmverlaufs die gesamte Zeile, die gerade bearbeitet wird, an.

TROFF
Syntax: TROFF oder TROFF Zeilenummer
Zweck: Schaltet die TRON-Funktion wieder aus.

TRON
Syntax: TRON oder TRON Zeilennummer oder TRON Zeilennummer-Zeilennummer
Zweck: Zeigt während des Programmverlaufs die Zeilennummer der Zeile an, die gerade bearbeitet wird, an.

UNBREAK
Syntax: UNBREAK oder UNBREAK Zeilennummer
Zweck: Widerruft einen BREAK-Befehl.

UNFOLLOW
Syntax: UNFOLLOW oder UNFOLLOW Variable,Variable
Zweck: Widerruft den FOLLOW-Befehl ganz oder für bestimmte Variablen.

UNTRACE
Syntax: UNTRACE oder UNTRACE Zeilennummer
Zweck: Widerruft den TRACE-Befehl.

USR
Syntax: X = USR Argument oder USRNummer Argument
Zweck: Ruft ein Maschinenunterprogramm mit einem Argument auf.

VAL
Syntax: X = VAL(Zeichenkette)
Zweck: Übergibt den numerischen Wert einer Zeichenkette.

VARPTR
Syntax: X = VARPTR(Variable) oder X = VARPTR(#Dateinummer)
Zweck: Überigbt die Arbeitsspeicheradresse einer Variablen oder eines Dateisteuerblocks.

VDISYS
Syntax: VDISYS(1)
Zweck: Erlaubt den Zugriff auf das VDI. Zuvor muß zum Schreiben in die Felder CONTRL, INTIN und PTSIN gePOKEt werden. Zum Lesen werden die Felder INTOUT und PTSOUT benutzt.

WAIT
Syntax: WAIT Datenkanalnummer,numerische Variable
Zweck: Hält den Programmverlauf an und kontrolliert den Status eines Ein-/Ausgabekanals.

WAVE
Syntax: WAVE Stimmenauswahl,Hüllkurve,Hüllkurvenzyklus,Dauer,Verzögerung
Zweck: Bestimmt die Hüllkurvenart für den SOUND-Befehl.

WEND
Syntax: WEND
Zweck: Stellt das Ende einer WHILE...Wend-Schleife dar.

WHILE
Syntax: WHILE logischer Ausdruck
Zweck: Führt eine Serie von Anweisungen in einer Schleife aus, so lange der logische Ausdruck richtig ist.

WIDTH
Syntax: WIDTH Breite oder WIDTH LPRINT Breite
Zweck: Setzt die Zeilenbreite auf Bildschirm oder Drucker.

WRITE
Syntax: WRITE Variable
Zweck: Gibt Daten auf dem Bildschirm aus.

WRITE#
Syntax: WRITE#Dateinummer,Variable
Zweck: Schreibt Daten in eine sequentielle Datei.

XOR
Syntax: Numerischer Wert XOR numerischer Wert
Zweck: Bildet die logische XOR-Verknüpfung (Antivalenz) zweier numerischer Werte.

Wahrheitstabelle:

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

&H
Syntax: X = &HHexadezimalzahl
Zweck: Wandelt eine Hexadezimalzahl in eine Dezimalzahl um.

&O
Syntax: X = &OOktalzahl
Zweck: Wandelt eine Oktalzahl in eine Dezimalzahl um.

Fehlernummern	Fehlermeldungen
2	Fehlerhafte Eingabe
3	RETURN-Anweisung erfordert entsprechendes GOSUB
4	READ-Anweisung hat zuwenig DATA-Werte
5	Funktionsaufruf nicht möglich
6	Zahl zu groß
7	Programm zu lang für Speicher
8	Anweisung oder Befehl bezieht sich auf eine nicht vorhandene Programmzeile
9	Unterdefinition bezieht sich auf ein Element außerhalb des Feldes
10	Feld mehr als einmal definiert
11	Division durch Null nicht möglich
12	Anweisung im Direkt-Modus ungültig
13	Wert-Arten stimmen nicht überein
15	String länger als 255 Zeichen
16	Ausdruck zu lang oder zu komplex
17	CONT arbeitet nur im BREAK-Modus
18	Funktion muß erst mit DEF FN definiert werden
20	RESUME-Anweisung vor Aufrufen der Fehler-Routine gefunden
22	Ausdruck beinhaltet Operator ohne nachfolgenden Operanden
23	Programmzeile zu lang
50	FIELD-Anweisung verursacht Überlauf
52	Ungültiger Dateiname oder -nummer
53	Datei im angegebenen Laufwerk nicht gefunden
54	Ungültiger Datei-Modus
55	Eine bereits geöffnete Datei kann nicht gelöscht (KILL) oder geöffnet (OPEN) werden
57	Diskettenein- oder -ausgabefehler. Operation neu einleiten (MP/M)
58	Datei existiert bereits
61	Diskette voll
62	Dateiende erreicht
63	Datensatz-Zahl in PUT oder GET ist größer als 32767 oder Null
64	Ungültiger Dateiname
65	Ungültiges Zeichen in der Programmdatei
66	Programmdatei enthält Anweisung ohne Zeilennummer
99	– Abbruch -
101	Programm für Speicher zu lang
102	ON-Anweisung außerhalb des gültigen Bereiches
103	Ungültige Zeilennummer
104	Variablen-Eingabe erforderlich
106	Zeilennummer existiert nicht
107	Zahl zu groß für eine Integer-Zahl
108	Eingegebene Daten ungültig. Eingabe ab dem ersten Begriff neu beginnen
109	STOP
110	Aufruf von Unterroutinen zu verschachtelt
111	Ungültige BLOAD-Datei
202	Befehl hier nicht möglich
203	Zeilennummernangabe erforderlich
204	FOR-Anweisung muß mit NEXT, oder WHILE-Anweisung mit WEND abgeschlossen werden
205	Einer NEXT-Anweisung muß FOR, oder einer WEND-Anweisung WHILE vorangestellt werden
206	Komma fehlt
207	Klammer fehlt
208	OPTION BASE muß 0 oder 1 sein
209	Abschluß der Anweisung erforderlich
210	Zu viele Argumente in der Liste
212	Redefinition von Variable(n) nicht möglich
213	Funktion mehr als einmal definiert
214	Sprung in eine Schleife nicht möglich
221	System-Fehler X, Neustart erforderlich
222	Programm läuft nicht
223	Zu viele FOR-Schleifen

Das neue ST-BASIC

ein BASIC der Superlative?

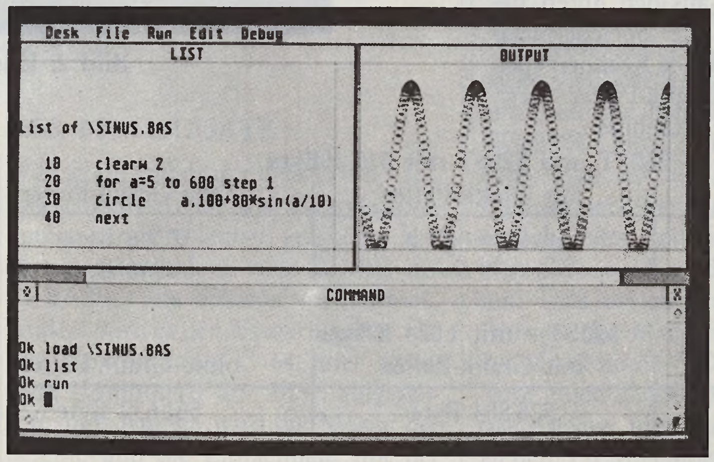


Bild 1: Die 4 Fenster des BASIC

Wenn man sich heutzutage einen Home- oder Personalcomputer kauft, gehört es normalerweise fast schon zur Grundausstattung, daß vom Hersteller ein BASIC-Interpreter mitgeliefert wird. Dies war wegen Programmanpassungsschwierigkeiten bei der ATARI-ST-Serie bisher noch nicht möglich. Es wurden zwar schon mehrere immer wieder auf den neusten Stand gebrachte Vorabversionen zur Verfügung gestellt. Aber das endgültige BASIC ließ auf sich warten. Die Endversion erreichte uns nun kurz vor Redaktionsschluß und wurde natürlich sofort einem kritischen Test unterzogen. Enttäuschung machte sich allerdings breit, als wir feststellten, daß diese Version mit der bereits Anfang Oktober erhältlichen Vorabversion identisch war. Damit war sofort klar, daß noch einige kleine Mängel vorhanden sein dürften. Doch auf diese wird noch ein paar Zeilen später tiefer eingegangen werden. Mitgeliefert werden soll ein ausführliches deutsches Handbuch, das sich zur Zeit allerdings noch im Druck befindet, aber sobald wie möglich nachgeliefert wird. Uns stand deswegen nur die Fotokopie des englischen Originalhandbuchs zur Verfügung.

Im Gegensatz zu verschiedenen Gerüchten, die besagten, daß das ST-BASIC zusammen mit dem TOS im internen ROM, also in einem Mikrochip untergebracht, in den ATARI ST implementiert werden sollte, wird es nun doch nur auf Diskette ausgeliefert. Es ist allerdings durchaus denkbar, daß es später einmal als externes Steckmodul für den Cartridge-Schacht erhältlich sein wird.

Mehr Speicherplatz

Die von uns getestete endgültige ST-BASIC-Version ist mit 138 944 Bytes deutlich kürzer als ihre beiden Vorgänger, die ca. 150 000 Bytes lang waren. Es bleibt allerdings das Speicherplatzproblem beim 260ST, dem nur 37 346 Bytes zur freien Verfügung bleiben. Hingegen der 520ST+ mit seinen 528 864 Bytes schon mehr als reichlich versorgt sein dürfte. Sobald jedoch das TOS endlich in ROM-Form ausgeliefert wird, wird auch der Mangel an Speicherplatz behoben werden, da dann die über 200 KBytes des Betriebssystems dem BASIC zur Verfügung stehen.

Wer allerdings jetzt schon mehr Speicherplatz benötigt, der kann sich mit ein paar Tricks leicht bis zu ca. 70 KBytes verschaffen. Hierbei kann man auf zwei verschiedene Methoden vorgehen.

Die erste Möglichkeit besteht darin, beim Laden des TOS von der Systemdiskette die DESK-Accessories wegzulassen. Dazu kopiert man am besten die Original-Systemdiskette (Die Erfahrung lehrt, daß es sich sowieso empfiehlt eine Sicherheitskopie anzulegen) und legt einen neuen Ordner an. In diesen Ordner kopiert man dann alle Dateien der Systemdiskette, die die Extension .ACC haben. Anschließend löscht man dieselben Dateien aus dem Hauptverzeichnis und behält somit eine Sicherheitskopie im Ordner zurück. Wenn man nun das System mittels des RESET-Knopfes neu startet, kann man feststellen, daß im Desktop die Einstellmöglichkeiten für die RS 232C-Schnittstelle, das Kontrollfeld und der VT52-Emulator fehlt. Dadurch gewinnt man über 20 KByte Speicher.

Die zweite Möglichkeit besteht darin, den Grafik-Puffer im BASIC auszuschalten. Dazu geht man in das RUN-Menü des BASICs und findet dort den Menüpunkt **BUF GRAPHICS.** Ist nun vor diesem Punkt ein Häkchen, ist der Grafik-Puffer eingeschaltet. In diesem Falle schaltet man den Puffer mittels der Maus durch ein einfaches Anklicken aus (Dies ist bei einer Speicherkapazität von 512 KByte und dazugeladenen DESK-Accessories nicht

mehr nötig, da der Puffer dann schon automatisch durch das BASIC ausgeschaltet wird). Der einzige Nachteil dieser Methode der Speicherplatzgewinnung ist, daß nun durch Fenster überlagerte Texte und Grafiken nicht mehr zwischengespeichert werden und somit leicht verloren gehen können. Wer einen 520ST+ hat kennt sowieso keine Speicherplatzprobleme mehr. Für alle anderen ST-Besitzer empfiehlt es sich auf Dauer, den Rechner selbst oder durch geübte „Lötkolbenartisten“ auf ein Megabyte aufzurüsten. Wer jedoch diese Möglichkeit scheut, kann aus den unten stehenden Tabellen die Speicherplatzmöglichkeiten für beide Rechnertypen mit und ohne Grafik-Puffer bzw. DESK-Accessories entnehmen.

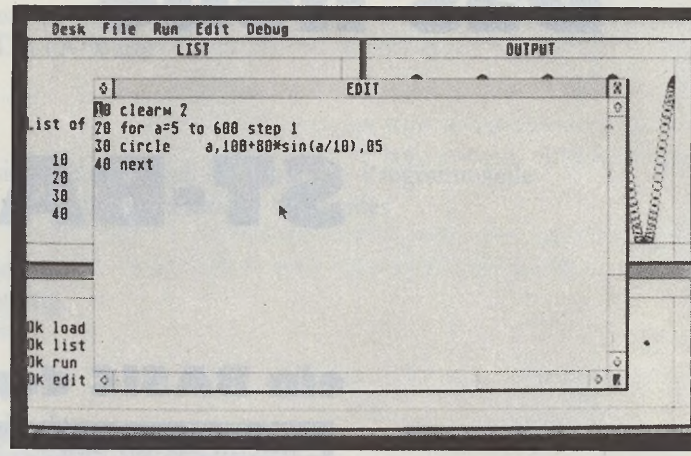


Bild 2: Das Edit-Fenster

260ST und 520ST mit 512 KByte

	mit Grafik-Puffer	ohne Grafik-Puffer
mit DESK-Acc.	nicht möglich	37 346 Byte
ohne DESK-Acc.	34 178 Byte	66 946 Byte

520ST+ mit 1 024 KByte

	mit Grafik-Puffer	ohne Grafik-Puffer
mit DESK-Acc.	528 864 Byte	561 632 Byte
ohne DESK-Acc.	558 464 Byte	591 232 Byte

Fenster im BASIC

Das ST-BASIC verfügt auf dem Bildschirm über vier verschiedene Fenster (siehe Bild 1), wobei das Vierte durch die drei Anderen überlagert ist. Die Fenster sind im einzelnen:

Das **LIST-Fenster**, in dem ein Programmlisting mittels des LIST-Befehls ausgegeben werden kann.

Das **OUTPUT-Fenster**, in dem sämtliche Ausgaben eines Programms erfolgen.

Das **COMMAND-Fenster**, das zur direkten Kommunikation mit dem BASIC-Interpreter dient.

Zu guter letzt das **EDIT-Fenster**, in dem Programme geschrieben und verändert werden können (siehe Bild 2). Dazu wäre noch zu sagen, daß man auch direkt im COMMAND-Fenster Programmzeilen eingeben kann. Dies bietet natürlich gute Programmiermöglichkeiten, da man alle vier Fenster gleichzeitig auf einen Blick darstellen kann.

Da das ST-BASIC voll GEM-unterstützt arbeitet, ist es auch möglich, mit der Maus jedes Fenster auf beliebige Größe zu bringen (siehe Bild 3). Man kann außerdem jedes Fenster gegenseitig überlagern oder wegklicken. Dieser GEM-Zugriff ist jedoch nicht nur über die Maus möglich, sondern es stehen dafür im BASIC-Befehlssatz die Befehle FULLW, CLEARW und CLOSEW zur Verfügung. Wie es unschwer zu erkennen ist, öffnet FULLW ein Fenster auf volle Bildschirmgröße. Des weiteren löscht CLEARW den Fensterinhalt vollständig. CLOSEW schließt ein Fenster. Um nun ein bestimmtes Fenster mit diesen Befehlen anzusteuern, sind ihnen Nummern zugeteilt worden, die man der folgenden Tabelle entnehmen kann.

EDIT-Fenster	=	0
LIST-Fenster	=	1
OUTPUT-Fenster	=	2
COMMAND-Fenster	=	3

Zusätzlich gibt es noch den **ED-** oder **EDIT**-Befehl, der das Edit-Fenster öffnet, sowie den **LIST**-Befehl, der ein Programmlisting im List-Fenster ausgibt. Diese beiden letzten Befehle werden allerdings nicht programmiert, sondern direkt im Command-Fenseter eingegeben.

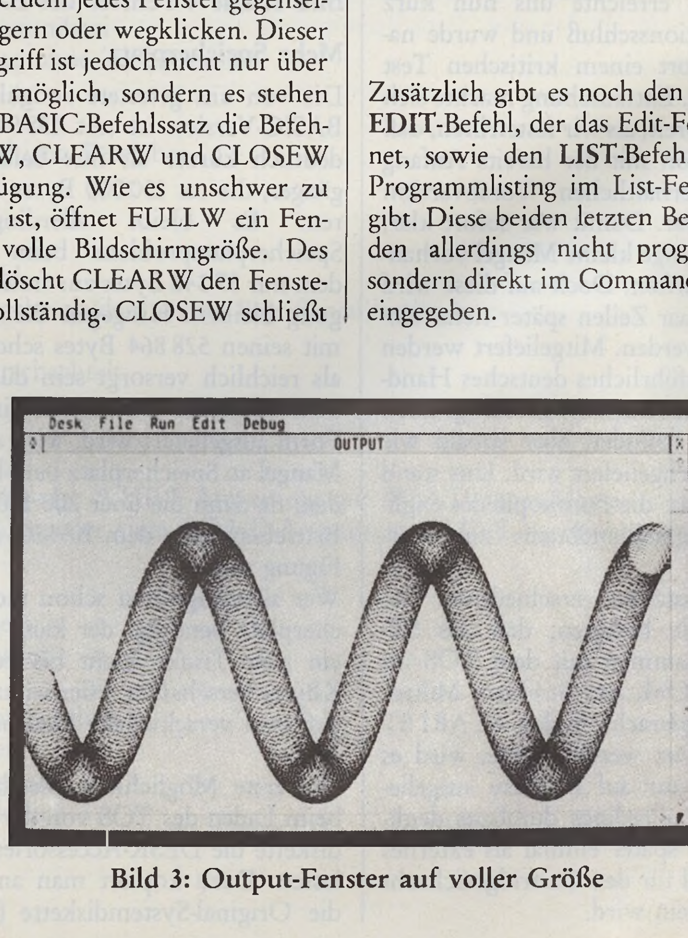


Bild 3: Output-Fenster auf voller Größe

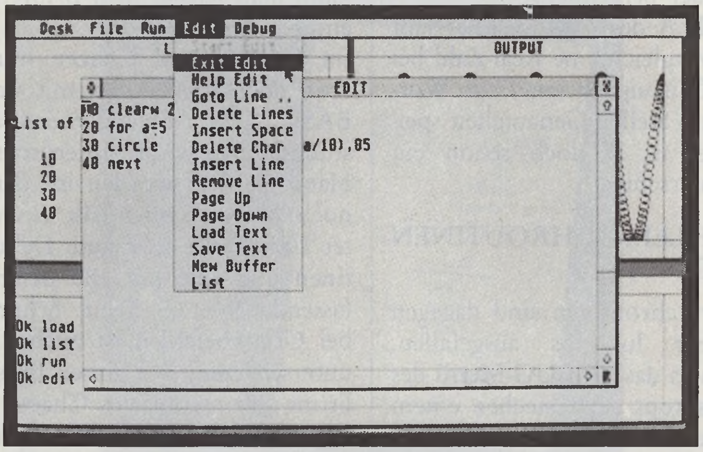


Bild 4: Das Edit-Menü

PULL-DOWN-MENÜS

Dem Anwender stehen im ST-BASIC fünf verschiedene Pull-Down-Menüs zur Verfügung. Sie enthalten folgende Menüpunkte:

Von links nach rechts gelesen steht an erster Stelle das **DESK-Menü,** das die BASIC-Copyright-Meldung und die DESK-Accessories enthält. Danach folgt das **FILE-Menü,** mit dem die wichtigsten Floppy-Operationen gesteuert werden können. An dritter Stelle steht dann das **RUN-Menü,** das die Programmstart- und stopbefehle, sowie den Einzelschrittmodus und die Grafikpufferung beinhaltet. Das folgende **EDIT-Menü** bietet ein reichhaltiges Sortiment an Editierfunktionen (siehe Bild 4), die aber fast alle auch über die zehn Funktionstasten abgerufen werden können. Die Funktionstastenbelegung sieht im einzelnen folgendermaßen aus:

F1 = Eine Leerstelle einfügen
F2 = Einen Buchstaben löschen
F3 = Eine Zeile einfügen
F4 = Eine Zeile löschen
F5 = Eine Bildschirmseite nach oben blättern
F6 = Eine Bildschirmseite nach unten blättern
F7 = Ein Programm laden
F8 = Ein Programm abspeichern
F9 = Einen neuen EDIT-Puffer bereitstellen
F10 = Den EDIT-Modus abschalten

Auf das letzte Menü, das **DEBUG-Menü** wird dann ein paar Zeilen tiefer bei den Fehlersuchroutinen noch genauer eingegangen.

DER BEFEHLSSATZ

Wer schon jemals mit dem GWBASIC bzw. BASICA, die auf Personalcomputern wie IBM laufen, gearbeitet hat, dem werden schon viele Befehle des ST-BASICs sehr bekannt vorkommen. Anders wird es den Käufern gehen, die von Commodores C 64 oder ähnlichen Computern um- bzw. aufgestiegen sind. Auf sie wird die Menge der Befehle und die Einfachheit, mit der man nun seine Programmierprobleme lösen kann, doch einigen Eindruck machen. Ein paar der Befehle mit ihren Möglichkeiten werden im weiteren kurz angesprochen.

Sehr viele gute Grafikmöglichkeiten bietet so zum Beispiel der **COLOR-**Befehl. Mit ihm kann man nicht nur die Text-, Füll- und Zeichenfarbe definieren, sondern auch noch mittels der beiden letzten Parameter auf die im Rechnerzeichensatz enthaltenen Füllmuster zugreifen. Ein deutliches Beispiel einer Anwendung dieses Befehls bietet das in diesem Heft enthaltene Grafikprogrammlisting. Wie die Muster im einzelnen abgerufen werden können, kann man dem Bild 5 genau entnehmen.

Natürlich noch ungewohnt ist die Arbeit mit dem GEM unter BASIC. Dazu dienen u. a. die oben schon besprochenen Fensterbefehle FULLW, CLEARW und CLOSEW. Doch genauso viel Beachtung fanden bei unserem Test die Befehle **GEMSYS** und **VDISYS.** Durch den GEMSYS-Befehl kann man auf die **AES**-(Application Environment Services) Kontrollfelder zurückgreifen und diese mit der Variablen **GB** und dem PEEK-Befehl auslesen. Dies erlaubt zum Beispiel eine Abfrage der Mauszeigerposition oder der Mausknöpfe. **VDISYS** dagegen erlaubt einen Zugriff auf das **VDI** (Virtual Device Interface). Im Gegensatz zu dem GEMSYS-Befehl kann bei Anwendung des VDISYS-Befehls auch in die Felder geschrieben werden. Dazu muß man einige vordefinierte Variablen benutzen. Zum Auslesen mit dem PEEK-Befehl lauten sie **INTOUT** und **PTSOUT,** zum Einlesen in den Speicher mit dem POKE-Befehl **CONTRL, INTIN** und **PTSIN.**

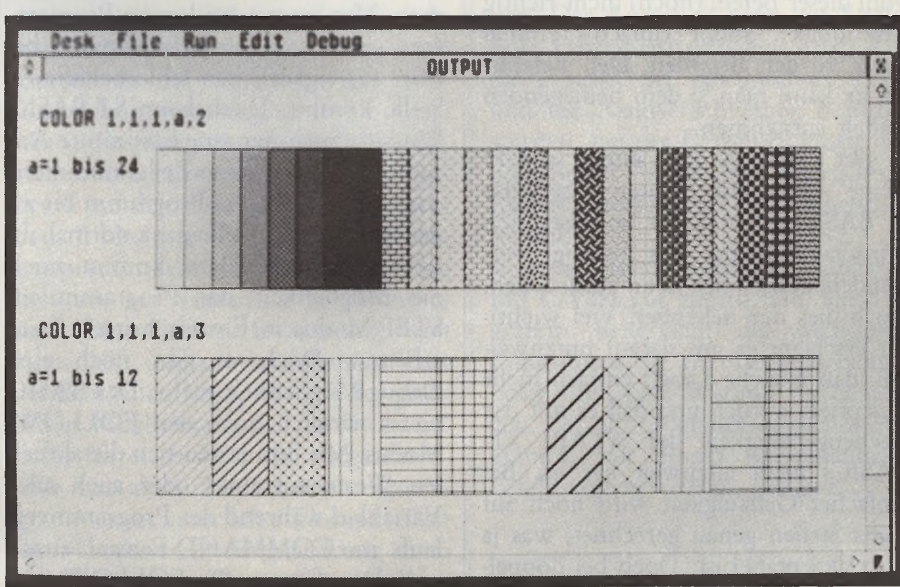


Bild 5: Die Füllmuster des BASICs

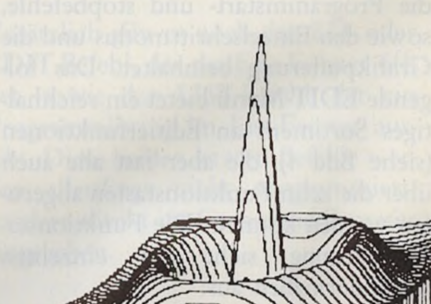
Weitere wichtige Befehle sind **INP** und **OUT**, die den Zugriff auf die einzelnen Schnittstellen ohne lästiges PEEK und POKE erlauben. Die Geräteadressen sind der folgenden Tabelle zu entnehmen:

Centronics-Schnittstelle	=	0
RS 232C-Schnittstelle	=	1
Tastatur und Bildschirm	=	2
MIDI-Schnittstelle	=	3
Tastaturport	=	4

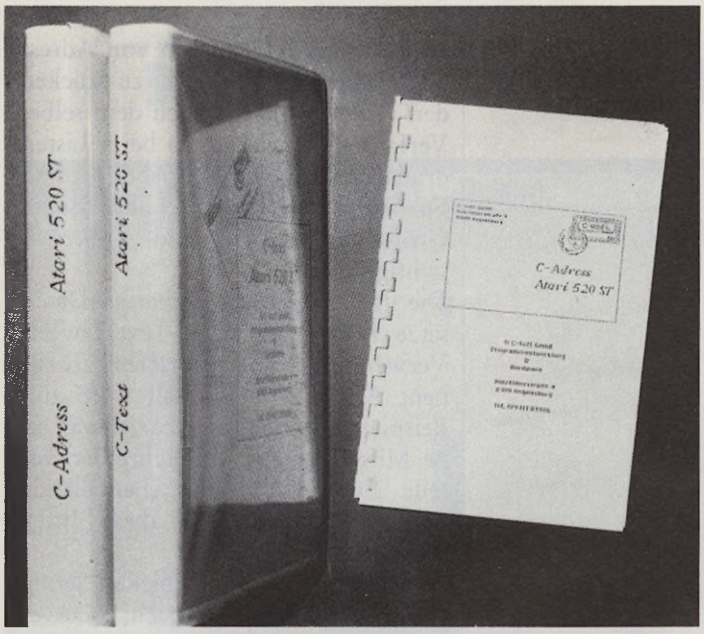
Auf jeden Fall muß man noch den **PEEK-** und den **POKE**-Befehl erwähnen, die bestimmt jedem BASIC-Programmierer von anderen Rechnern her geläufig sind. Beim ST BASIC gibt es allerdings einen wichtigen Unterschied zu den anderen BASIC-Versionen. PEEK und POKE lesen bzw. schreiben nämlich ein ganzes Wort (zwei Speicherzellen), was einer 16-Bit Zahl entspricht. Man muß also, wenn man nur eine Speicherzelle auslesen will, die andere mit einer logischen AND-Verknüpfung ausblenden.
Doch nicht nur Positivies ist uns bei den Befehlen aufgefallen. So arbeitet der **GOTOXY**-Befehl leider immer noch nicht richtig. Er läßt sich zwar in Programmen verwenden, doch wenn man X um eins erhöht, gibt er stattdessen auf dem Bildschirm eine Erhöhung um zwei aus. Dieser Befehl ist also mit Vorsicht zu genießen. Leider kann man ihn aber nur durch lästige TABs und SPCs umgehen. In Anbetracht dieser Tatsache ist es schon verwunderlich, daß ATARI dieses ST-BASIC als Endversion herausgibt, obwohl dieser Befehl (noch) nicht richtig funktioniert. Somit zunächst einmal genug zu den Befehlen. Den Befehlssyntax kann man ja dem beiliegenden Beiheft entnehmen.
Es gibt allerdings noch einige weitere kleine Unerfreulichkeiten über das ST-BASIC zu berichten. So lassen sich deutsche Umlaute nur bedingt verwenden, geschweige denn ausdrucken. Doch dies nur nebenbei. Viel wichtiger erscheint es uns darauf hinzuweisen, daß manche Labels einfach nicht akzeptiert werden und daß es mit der Rechengenauigkeit des ATARI ST-BASICs nicht allzuweit her ist. Bei einfacher Genauigkeit wird noch auf sechs Stellen genau gerechnet, was ja ganz akzeptabel ist. Doch bei doppelter Genauigkeit läßt sich das Ergebnis einer Rechenoperation nur um eine Stelle genauer berechnen.
Beim BASICA der IBM-Rechner läßt sich zum Vergleich eine Real-Zahl bei doppelter Genauigkeit mit 17 (in Worten siebzehn) Stellen Genauigkeit speichern. Dies ist ja doch schon ein kleiner Unterschied.

DIE FEHLERSUCHROUTINEN (DEBUGGING)

Die Fehlersuchroutinen sind dagegen wieder sehr luxuriös ausgefallen. Wenn man in das **DEBUG-Menü** des BASIC-Desktops geht, stechen einem sogleich die Befehle **TRON, TROFF, TRACE** und **UNTRACE** ins Auge. Mit ihnen werden die entsprechenden Modi ein- bzw. ausgeschaltet und der erfahrene BASIC-Programmierer weiß sofort, daß sie zur Fehlersuche, dem Debugging, dienen. Das einzige, was einen stutzig macht, ist, daß TRON die Abkürzung für TRACE ON und TROFF für TRACE OFF ist. Was sollen also zwei TRACE-Befehle? Die Praxis zeigt es sofort. Der TRON-Modus zeigt während eines Programmablaufs nur die aktuellen Zeilennummern im COMMAND-Fenster an. TRACE dagegen liefert die komplette aktuelle Programmzeile. Außerdem bietet das ST-BASIC noch die Möglichkeit diese beiden Debug-Modi auf bestimmte Zeilen zu beschränken, was von ganz entscheidendem Vorteil ist. Denn wer jemals ein Programm mit TRACE bzw. TRON abarbeiten ließ, der weiß auch, daß diese beiden Modi den Programmverlauf stark verlangsamen. Man wartet bei langen Programmen schon mal ein paar Minuten bis man zur eigentlichen fehlerbehafteten Stelle kommt. Doch beim ST-BASIC läßt sich auch nur eine bestimmte Anzahl von Zeilen bzw. Zeilennummern anzeigen, so daß das Programm bis zu der bestimmten Stelle ganz normal abgearbeitet wird. Dazu kommt noch die Möglichkeit das Programm im STEP-Modus in Einzelschritten abzuarbeiten. Doch es gibt noch eine Debug-Möglichkeit mehr. Das BASIC bietet nämlich noch den **FOLLOW**-Modus. Mit ihm lassen sich die aktuellen Werte einzelner oder auch aller Variablen während des Programmverlaufs im COMMAND-Fenster anzeigen. Den Syntax für FOLLOW und seinem Gegenstück UNFOLLOW kann man wieder dem BASIC-Beiheft entnehmen.
Im Großen und Ganzen muß man aber doch sagen, daß mit dem ST-BASIC dem Anwender ein sehr leistungsfähiger BASIC-Interpreter in die Hand gegeben worden ist, der kaum noch Wünsche offen läßt. Er verarbeitet Labels, hat sehr gute Debug-Routinen und eine fast 180 Befehle umfassende Syntax. Seine Schnelligkeit bei Grafikbefehlen ist auch nicht zu unterschätzen, wie unser Programmlisting „Kontrolliertes Chaos“ in diesem Heft beweist. Nicht zu vergessen, daß er diverse logische Verknüpfungen beherrscht. Es bleiben da zwar noch die negativen Eigenschaften, aber die lassen sich bestimmt mittels eigener Maschinensprachroutinen, die man ja mit **CALL** oder **USR** aufrufen kann, beheben. Außerdem bleibt noch abzuwarten, ob die Softwarefirmen Erweiterungen für das ST-BASIC auf den Markt bringen. Nicht zuletzt kann man noch auf die wertvolle Arbeit von ST-Anwendern und -„Freaks“ hoffen, die schon jetzt teilweise recht interessante Programme geschrieben haben.



Für diese Grafik benötigte das BASIC ca. 120 Sekunden



C-Text & C-Adress

Diese zwei Programme, von der Firma C-Soft, werden zusammen als kombiniertes Textverarbeitungs- & Adressverwaltungs-Paket angeboten. Dadurch hat man die Möglichkeit beim Schreiben von Texten auf eine bestehende Adressdatei zurückzugreifen.
C-Text (Version 1.0), das Textverarbeitungsprogramm, hat ähnliche Möglichkeiten wie Wordstar, das weitverbreiteste Textprogramm überhaupt.
Das Programm, das mit dem Monochrom- und dem Farbmonitor arbeitet, verfügt über alle Standardtextbearbeitungsbefehle, die zur Texterstellung nötig sind. So gibt es unter anderem die Möglichkeit ganze Textabschnitte zu verschieben, zu kopieren oder zu löschen (Blockkommandos). Dazu markiert man einen Block beliebiger Größe und wählt dann über eine Tastenkombination die gewünschte Funktion an.
In der ersten Bildschirmzeile werden Informationen über die Cursorposition, Bearbeitungsmodus und eventuelle Fehlermeldungen angezeigt.
Alle Befehle sind über die Control-Taste erreichbar. Hier macht sich die Übereinstimmung mit Wordstar positiv bemerkbar. Der ein oder andere ist sicher mit diesem Programm schon in Berührung gekommen und ist diese Tastenbelegung deshalb gewöhnt. Auch in vielen bekannten anderen Programmen wie z. B. Turbo Pascal, werden die gleichen Control-Tasten zu dem gleichen Zweck benutzt.
Die wichtigsten Bedienungsbefehle kann man ebenso mit Hilfe der Funktionstasten aufrufen. Dabei stehen dem Benutzer allerdings nur neun Tasten zur Verfügung. Eine Mehrfachbelegung der Funktionstasten, beispielsweise in der Form von anwählbaren Untermenüs, hätte hier eine zusätzliche Bedienungserleichterung geschaffen.
Neben den normalen Cursorfunktionen, die natürlich auch über die Cursortasten erreichbar sind, gibt es noch die Möglichkeit wortweise und an den Anfang bzw. das Ende einer Zeile zu springen. Ebenso besteht die Möglichkeit seitenweise zu blättern und den Cursor an das Textende- bzw. den Textanfang zu bewegen.
Mit C-Text kann man bestimmte Textstellen unterstreichen oder im Fettdruck auf dem Drucker ausgeben. Auf dem Bildschirm wird dies mit einem entsprechendem Grafikzeichen, ähnlich wie bei Vizawrite, gekennzeichnet. Die direkte Darstellung von Fettschrift und unterstrichenem Text auf dem Bildschirm, ist trotz der grafischen Möglichkeit, die der Atari ST dafür bietet, leider nicht vorgesehen.
Wie bei allen Textverarbeitungsprogrammen ist natürlich auch bei C-Text die Fähigkeit zur Formatierung eines Textes gegeben. Damit können überflüssige Leerstellen die beim Löschen oder Verschieben von Textteilen entstehen, entfernt werden.
Zum Drucken kann man Drucker verwenden, die entweder mit ASCII-Zeichen angesteuert werden, oder IBM-Drucker kompatibel sind. Darunter fallen unter anderem alle Epson- und Epson kompatible Drucker.
Die Druckersteuerzeichen für Unterstreichen, Fettdruck und 'Near Letter Quality' sind an den benutzten Drucker angleichbar. Diese Daten werden von dem Programm selbstverständlich auf Diskette abgelegt, sodaß sie bei jedem Starten des Textsystems automatisch mitgeladen werden können. Der Ausdruck kann auf Endlos- oder Einzelblatt erfolgen, dabei kann die Anzahl der gewünschten Zeilen und die Randverschiebung bestimmt werden. Bei der Einzelblatteinstellung unterbricht C-Text den Ausdruck und fordert zum Einlegen eines neuen Blattes auf.
Insgesamt ist nach einiger Übungszeit, mit C-Text gut zu arbeiten. Das Programm beherrscht die wichtigsten Funktionen die zur Textverarbeitung nötig sind, und ist somit für denjenigen, der ein möglichst einfaches Programm, mit einem überschaubaren Befehlssatz bevorzugt, geeignet.

Befehlsübersicht C-Text:

Cursorsteuerung:
- Zeichen links / rechts
- Zeichen hoch / runter
- Wort links / rechts
- Zeilenende / Zeilenanfang
- Zeile vor / zurück
- Seite vor / zurück
- Textanfang / Textende

Löschen und Einfügen:
- Zeichen löschen
- Wort löschen
- Zeile löschen
- Leerzeile einfügen
- Text einfügen

Blockbefehle:
- Blockmarkierung
- Block löschen
- Block verschieben
- Block copieren

Druckgestaltung:
- Fettdruck
- Unterstreichen
- Einzelblattverarbeitung
- Seitennummerierung
- Randverschiebung (links)

Datenübernahme:
- Mailmerge mit C-Text
- Texteinlesen ab Cursorposition

Sonstiges:
- Textformatieren
- Markierungen setzen u. anspringen

C-Adress

Es handelt sich hierbei um eine einfache Adressverwaltung, die im privaten Bereich und bei der Verwaltung eines kleineren Unternehmens ihre Anwendung findet.

Bei größeren Datenmengen geht in einem normalen Adressbuch die Übersicht leicht verloren, da hier nur nach einem Kriterium geordnet wird. Mit einem Adressverwaltungsprogramm ist es nun möglich nach verschiedenen Kriterien zu ordnen und gezielt auf Daten zuzugreifen.

Bei C-Adress sind zwei Sortierkriterien implementiert, nämlich wahlweise nach Namen oder Kundennummern. Die gesamten Daten dieser Dateien können als kundenspezifische Stammblätter ausgegeben werden oder als Teilauszüge in Listen ihre Verwendung finden. Diese Listen lassen sich nach vielfältigen Auswahlpunkten zusammenstellen. Denkbar wäre eine Telefonliste, in der nur die Namen und die Telefonnummern enthalten ist.

Bei der Auswahl der gewünschten Adressen kann man nun eine bestimmte Personengruppe selektieren, so zum Beispiel Personen, deren Name mit F...K beginnt und die in einer bestimmten Stadt wohnen.

Diese so gewonnenen Datensätze können auf dem Drucker ausgegeben werden, wobei das Format dieser Liste frei wählbar ist. Man erhält dabei eine übersichtliche Zusammenstellung der gewünschten Daten, die je nach Bedarf mit verschieden breiten Schrifttypen gedruckt werden kann. C-Adress unterstützt alle IBM und Epson kompatiblen Drucker, sowie die Drucker die Kodes nach der ASCII-Norm annehmen.

Ein weiteres Anwendungsgebiet des C-Adress ist das Anfertigen von Adressetiketten. Die Auswahl der zu druckenden Daten geschieht nach dem selben Verfahren, welches auch beim Listendruck verwendet wird. Dabei wird das Format der Etiketten – ihre Breite, Größe und Randabstand – berücksichtigt.

Die in C-Adress gespeicherten Datensätze kann man an C-Text, zwecks Verwendung in Serienbriefen, übergeben. Als Anwendungsfall wäre zum Beispiel ein Verein denkbar, der an seine Mitglieder ein Rundschreiben verteilt. Dazu kann man ebenfalls die Anschriftaufkleber für diese Briefe drucken lassen.

Die Anfertigung eines solchen Serienbriefes ist denkbar einfach. Man erstellt dazu den Text und markiert darin die Stellen, an denen die Daten aus der Adressdatei automatisch eingefügt werden. Auf diese Art und Weise kann man die komplette Anschrift, die Anrede oder sonstige Informationen aus der Adressdatei entnehmen.

Die Arbeitserleichterung, die durch diese Programmfunktion bewirkt wird, ist eine der interessantesten Fähigkeiten dieses Programmes.

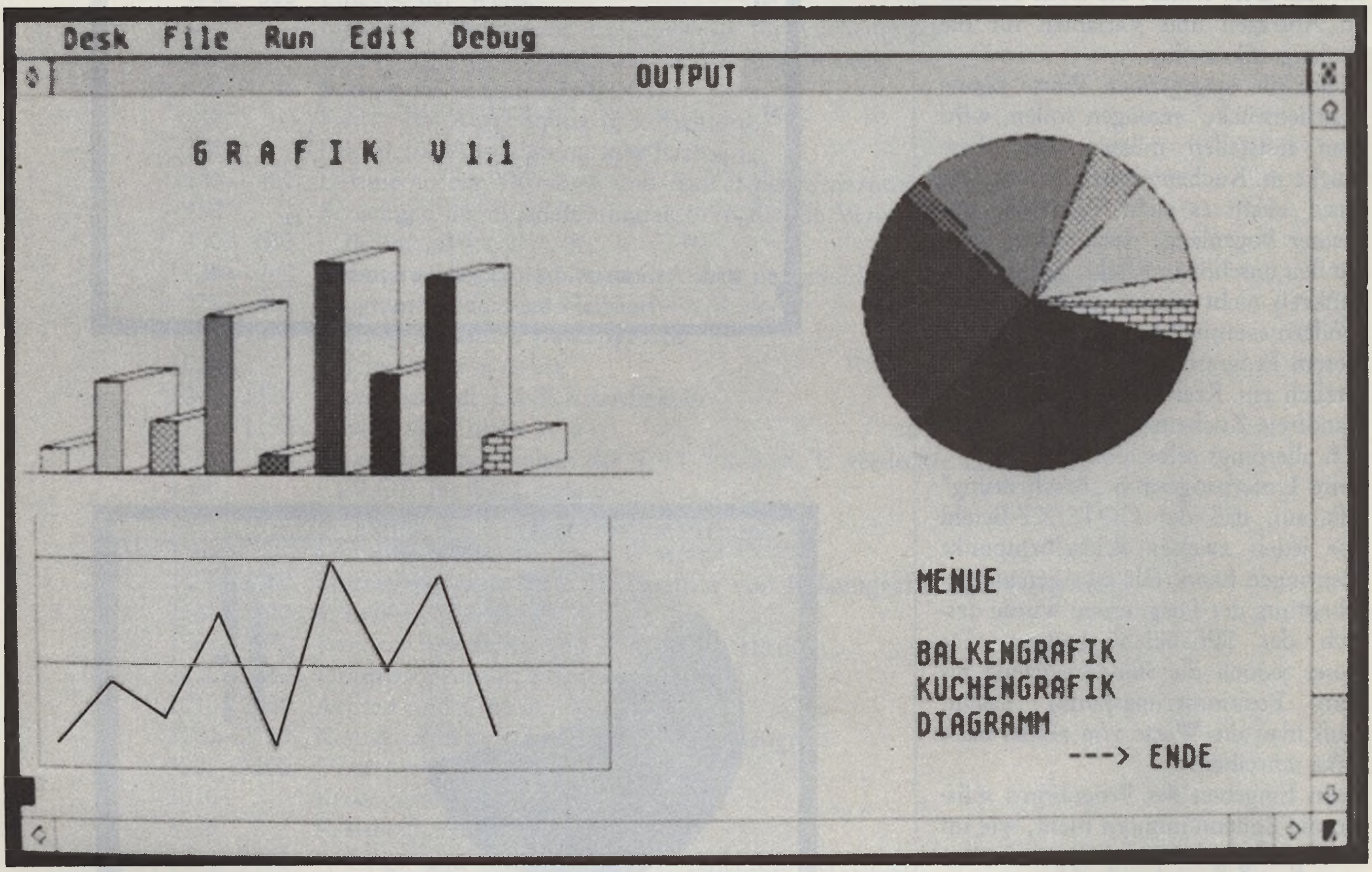
C-Adress ist im Allgemeinen einfach gehalten und bereitet keine Schwierigkeiten bei der Bedienung. Das Handbuch erklärt in einer kurzen Form die Programmfunktionen die zur Bedienung nötig werden, sollte aber für den unerfahrenen Anwender weitere Beispiele enthalten.

Der Benutzer kann auf eine vordefinierte Maske zurückgreifen und kann somit sofort mit der Adressbearbeitung beginnen.

Bei der Listenerstellung und bei den Etiketten kann man aus dieser Dateimaske auswählen welche Informationen entnommen werden sollen sowie deren Länge. Bei der Datenmaske ist man allerdings sehr eingeschränkt. Man muß diese festgelegte Maske verwenden, da eine Selbsterstellung nicht möglich ist. Ebenso kann man die Datensatzlänge und die Beschriftung der Felder nicht auf die persönlichen Wünsche anpassen.

Von diesen Punkten abgesehen, ist C-Adress ein hilfreiches Programm, das so manche Such- und Sortierarbeit übernimmt und darüber hinaus in Zusammenarbeit mit C-Text zur komfortablen Serienbrieferstellung geeignet ist.

Kuchen- & Säulengrafik V 1.0



Grafische Darstellungsformen

Da normale Zahlentabellen meist unanschaulich und schwer zu verarbeiten sind, gibt es die Möglichkeit, sie grafisch umzusetzen. So kann man z. B. die Werte im normalen zweidimensionalen Koordinatensystem darstellen, wie es bei mathematischen Funktionen üblich ist. Im Gegensatz zu reinen Zahlenwerten kann man anhand eines solchen Diagrammes das Verhältnis der Werte zueinander wesentlich besser erfassen. Ein weiterer Vorteil dieser Darstellungsart liegt im sofortigen Erkennen von Steigungsveränderungen. Dies ist zum Beispiel bei der Auswertung von Meßergebnissen wichtig oder bei der Analyse von Verkaufszahlen.

Eine optisch ansprechendere Darstellungsweise erhält man mit dem Balkendiagramm. Hierbei hat man die Möglichkeit, die Werte durch verschiedene Farben oder Schraffuren zu unterscheiden oder Gemeinsamkeiten hervorzuheben.

Eine erweiterte Form dieser Grafikart ist die Säulengrafik. Der Unterschied zur Balkengrafik liegt darin, daß hier eine Räumlichkeit durch die dreidimensionale Darstellung erzeugt wird. Hiermit können auch mehrspaltige Säulenreihen erstellt werden wie es bei hochwertigen Programmpaketen (z.B. OPEN ACCESS) realisiert ist.

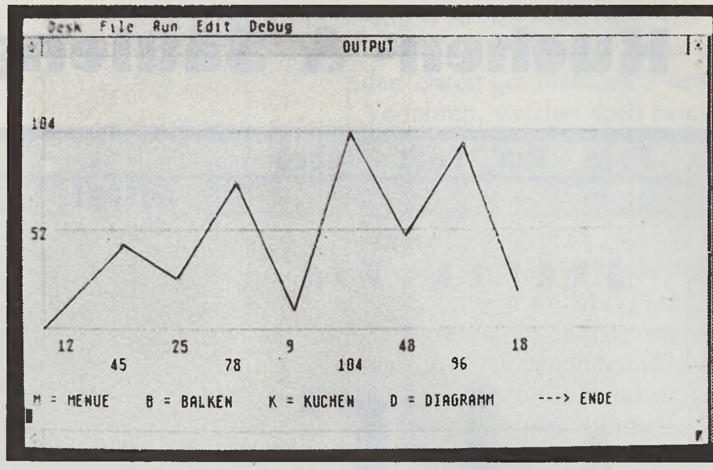
Auch die Kuchengrafik ist eine oft verwendete Form der grafischen Darstellung. Sie liefert übersichtliche und vor allem kompakte Bilder und wird gerne zur prozentualen Darstellung benutzt.

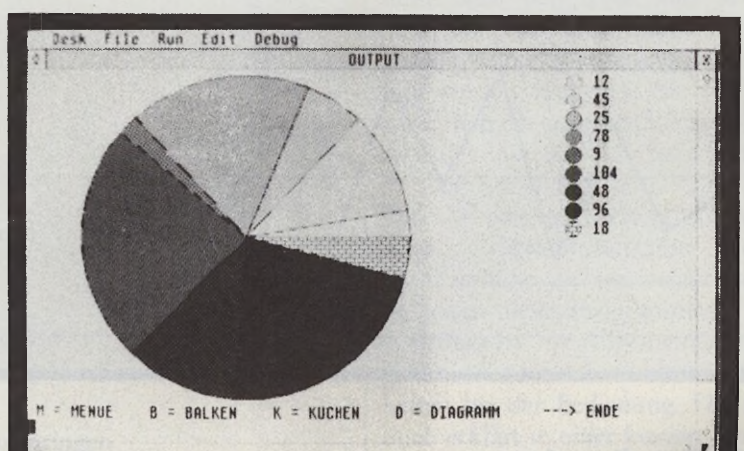
Doch nun noch einige Worte zum Programm selbst. Das Programm „Kuchen- & Säulengrafik" gibt die eingegebenen Werte als Säulen- und Kuchengrafik und als Diagramm aus. Bei der Eingabe ist zu beachten, daß das Programm momentan nur 18 Werte annimmt und dann automatisch mit der Berechnung beginnt; dies kann natürlich geändert werden. Die Eingabe kann man jederzeit mit „★" beenden. Das Programm stellt dann das Gesamtbild zusammen. Es werden dabei alle Grafikarten, jedoch ohne Werte, dargestellt. Zusätzlich erscheint das Menü, mit dem man nun jede beliebige Grafik auf den gesamten Bildschirm vergrößern kann. Zu dieser Vergrößerung werden auch die Werte ausgegeben. Zwischen den Grafiken und der Gesamtdarstellung kann nun beliebig umgeschaltet werden. Dazu wird nur der entsprechende Anfangsbuchstabe gedrückt.

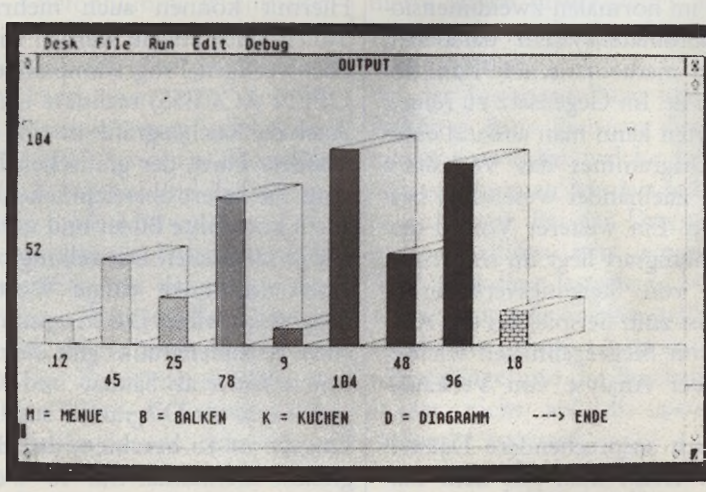
Bei dieser Eingabe werden Groß- und Kleinbuchstaben vom Programm abgefragt, da der INPUT$-Befehl eine Unterscheidung zwischen Groß- und Kleinschrift macht und somit bei versehentlichem Drücken der CAPS-LOCK Taste nicht mehr reagiert.

Das Unterprogramm zum Zeichnen der Säulen ist etwas kompliziert geworden. Dies liegt an der gewählten Darstellungsweise, bei der sich die Säulen zum Teil überschneiden. Es ist deshalb notwendig, die unter der Nebensäule liegenden Linien wegfallen zu

lassen. Da es viele verschiedene Überlagerungssituationen gibt, ist es notwendig jede Linie getrennt zu untersuchen. Dies erklärt die hohe Anzahl an Abfragen und Variablen für die Säulengrafikroutine.
Wenn die eingegebnen Werte kleine 'Kuchenstücke' erzeugen sollen, wird man feststellen müssen, daß diese Stücke im Kuchen einfach fehlen. Das Basic schafft es nicht Teilstücke mit kleiner Bogenlänge darzustellen. Dies hat den unschönen Effekt, daß der Außenkreis nicht immer geschlossen ist, sondern eventuell Lücken aufweist. In diesem Programm wurde deshalb zusätzlich ein Kreis gezogen. Eine einwandfreie Kuchenroutine müßte man sich allerdings selbst erstellen.
Beim Unterprogramm „Beschriftung" fällt auf, daß der GOTOXY-Befehl nur jeden zweiten Bildschirmpunkt anspringen kann. Für eine genaue Beschriftung der Diagramme wurde deshalb der SPC-Befehl benutzt. Da dieser jedoch die Bildschirmzeile bis zum Positionierungspunkt löscht, muß man die Werte von rechts nach links schreiben.
Beim Eingeben des Programms müssen die Zeilennummern nicht, wie im Listing angegeben, abgetippt werden. Sie können beliebig gewählt werden, weil das Programm keine Sprünge zu Zeilennummern enthält. Zum besseren Verständnis des Listings wurde ausschließlich mit Labels gearbeitet, die im Klartext ausdrücken wohin der Sprung erfolgt. Somit entfällt ein umständliches Suchen nach Sinn und Auswirkung des Sprunges. Das Programm ist so angelegt, daß es leicht den eigenen Anforderungen angepaßt werden kann. Das Listing besitzt eine übersichtliche TOP-DOWN-Struktur, von der aus die verschiedenen Unterprogramme aufgerufen werden. Als Unterprogramme sind die Zeichenroutinen für die verschiedenen Grafiken und die Beschriftungen angelegt. Fast alle in diesen Routinen verwendeten Koordinaten sind als Parameter definiert und somit leicht abzuändern. Die Parameter werden vor dem Aufruf des jeweiligen Unterprogramms definiert.







Programmdokumentation

100 – 170	Bildschirm vorbereiten, Dimensionierung
170 – 300	Eingabe der Werte
270	Bilden der Summe der Eingabewerte für Kuchengrafik
280	Ermitteln des größten Elements der Eingabewerte
320	Beginn des Hauptprogramms
340	Faktor für Ausdehnung in x-Richtung
350	Faktor für Ausdehnung in y-Richtung
380 – 430	Definieren der Variablen und Aufruf der Unterprogramme
450	Berechnen der Ausdehnungsfaktoren für die Vergrößerung
460 – 600	Hauptmenü
620 – 720	Definieren der Variablen und Aufruf des jeweiligen Unterprogramms
770 – 860	Programmende oder Neustart
1000 – 1370	**Unterprogramm: Balkengrafik**
1020	Zeichnen der x-Achse
1040 – 1120	Berechnen der LINE-Koordinaten
1130 – 1150	Ziehen der Linien 1 – 3
1160 – 1320	Berechnen und Ziehen der 4. – 7. Linie (z. T. verdeckte Linien)
1330	COLOR für FILL-Befehl
1340	Ausfüllen der Vorderseite der Säule
1360	COLOR-Befehl auf 'normal' setzen (!)
1500 – 1580	**Unterprogramm zum Beschriften von Balkengrafik und Diagramm**
1510 – 1520	Y-Achse beschriften
1530 – 1570	Position berechnen und X-Achse beschriften
2000 – 2140	**Unterprogramm: Diagramm**
2010 – 2050	Achsen und Einteilungslinien ziehen
2060 – 2120	Koordinaten berechnen und Linien ziehen
3000 – 3100	**Unterprogramm: Kuchengrafik**
3010	Kreis ziehen
3020	Kreisbogenfaktor berechnen
3040 – 3050	Anfangs- und Endwinkel des Kreissektors berechnen
3060 – 3070	Kreissektor zeichnen und ausfüllen
3090	COLOR-Befehl zurücksetzen (!)
3500 – 3570	**Unterprogramm zum Kennzeichnen der Kreissektoren**
3520	COLOR-Befehl setzen
3530	Beschriftungskreise zeichnen und ausfüllen
3540	Position berechnen und beschriften

VARIABLENLISTE

W()	Feld für Eingabewerte
E$	allgemeiner Eingabestring
A	allgemeine Schleifenvariable
NR	Anzahl der Eingabewerte
SU	Summe aller Eingabewerte
YM	größter y-Wert
XY	Abstand zum nächsten Wert (x-Achse)
YA	Vergrößerungsfaktor y-Achse
BX1,BX2,BY1	Koordinaten für x,y-Achse
XB	Abstand der Säulen (!)
XB/2	Breite der Säule
YB	Faktor für Lage der z-Achse
X1,X2,X3,X4	Koordinaten für Balken
Y1,Y2,Y3,Y4,Y5	(siehe Zeichnung)
DX1,DX2,DY1,DYY	Koordinaten für x,y-Achse
X1,Y1,X2,Y2	Koordinaten für LINE-Befehl
CX1,CY1,R1	Koordinaten für Kreismittelpunkt und Radius
W1	Kreisbogenfaktor
WA,WE	Anfangs- und Endwinkel

```
List of \GRAPH.BAS

 100  '      G R A F I K   V 1.0
 110  '            by mark
 120  '
 130  programmstart :
 140  CLEAR:DIM W(30)
 150  A$=" M = MENUE     B = BALKEN     K = KUCHEN     D = DIAGRAMM"
 160  A$=A$+"     ---> ENDE"
 170  COLOR 1,0,1 :FULLW 2: CLEARW 2 : CLOSEW 3
 180  GOTOXY 10,0:PRINT "GRAPHISCHE DARSTELLUNG"
 190  GOTOXY 5,2: PRINT"... bitte die Werte eingeben !  ENDE = :*:"
 200  GOTOXY ,5:NR=0
 210  '-----------------------------------------------------------
 220  eingabe :
 230  WHILE E$<>"*" AND NR <18
 240  NR=NR+1
 250  PRINT "            WERT ("NR ")= ";: INPUT E$
 260  W(NR)=ABS( VAL(E$))
 270  SU=SU+W(NR)
 280  IF YM<W(NR) THEN YM=W(NR)
 290  WEND
 300  IF E$="*" THEN NR=NR-1
 310  '-----------------------------------------------------------
 320  gesamtgrafik :
 330  CLEARW 2
 340  XA=INT(240/NR)
 350  YA=100/YM
 370  GOTOXY 5,1: PRINT"G R A F I K   V 1.0"
 380  BX1= 5:BX2=300:BY1=180
 390  GOSUB balkengrafik
 400  DX1=10:DX2=280:DY1=320:DYY=200
 410  GOSUB diagramm
 420  CX1=480:CY1=100:R1=80
 430  GOSUB kuchengrafik
 440  '   ausdehnung
 450  XA=2*XA:YA=1.8*YA
 460  '   menue
 470  GOTOXY 25,13: PRINT "MENUE"
 480  GOTOXY 25,15: PRINT "BALKENGRAFIK"
 490  GOTOXY 25,16: PRINT "KUCHENGRAFIK"
 500  GOTOXY 25,17: PRINT "DIAGRAMM"
 510  GOTOXY 25,18: PRINT "          ---> ENDE"
 520  '-----------------------------------------------------------
 530  menue :
 540  E$=INPUT$(1)
 550  IF E$="M" OR E$="m" THEN GOTO gesamtgrafik
 560  IF E$="B" OR E$="b" THEN GOTO b
 570  IF E$="K" OR E$="k" THEN GOTO k
 580  IF E$="D" OR E$="d" THEN GOTO d
 590  IF E$="E" OR E$="e" THEN GOTO ende
 600  GOTO menue
```

List of \GRAPH.BAS

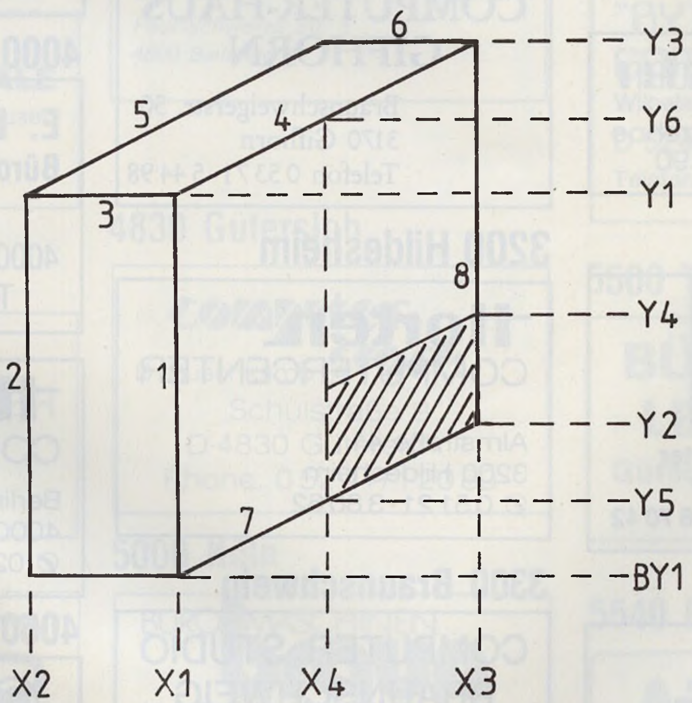
```
610   '
620   b: BX1=10:BX2=640:BY1=250
630   CLEARW 2
640   GOSUB balkengrafik  : GOSUB beschriftung : GOTO auswahl
650   '
660   k: CX1=200:CY1=155:R1=150
670   CLEARW 2
680   GOSUB kuchengrafik :GOSUB kuchenbeschriftung :GOTO auswahl
690   '
700   d: DX1=20:DX2=560:DY1=250:DYY=10
710   CLEARW 2
720   GOSUB diagramm : GOSUB beschriftung
730   '
740   auswahl :
750   GOTOXY 0,18: PRINT A$ : GOTO menue
760   '
770   ende :
780   GOTOXY 5,5:PRINT"NEUE DATEN"
790   GOTOXY 5,7:PRINT" ---> ENDE"
800   '
810   menue1 :
820   E$=INPUT$(1)
830   IF E$="N" OR E$="n" THEN GOTO programmstart
840   IF E$="E" OR E$="e" THEN END
850   GOTO menue1
860   END
870   '-----------------------------------------
1000  balkengrafik :
1010  COLOR 1,1,1,0,1
1020  LINEF BX1,BY1,BX2,BY1
1030  FOR A=NR TO 1 STEP -1
1040  XB=XA: YB=4
1050  X1=A*XA
1060  X2=X1-XB/2
1070  X3=X1+XB
1080  X4=X1+XB/2
1090  Y1=BY1-W(A)*YA
1100  Y2=BY1-XA/YB
1110  Y3=Y1-XA/YB
1120  Y5=BY1-XB/(YB*2)
1130  LINEF X1,BY1,X1,Y1         : REM 1
1140  LINEF X2,BY1,X2,Y1         : REM 2
1150  LINEF X1,Y1,X2,Y1          : REM 3
1160  Y6=Y5-W(A)*YA: X44=X4
1170  IF W(A)*YA>=W(A+1)*YA-XB/8 THEN X44=X3: Y6=Y3
1180  LINEF X1,Y1,X44,Y6         : REM 4
1190  LINEF X2,Y1,X2+XB,Y3       : REM 5
1200  IF Y3>BY1-W(A+1)*YA THEN GOTO noline
1210  X5=X4+4*(BY1-W(A+1)*YA-Y3)
1220  IF X5>X3 THEN X5=X3
```

List of \GRAPH.BAS

```
1230  LINEF X4,Y3,X5,Y3          : REM 6
1240  noline: IF(A=NR) THEN X4=X3:Y5=Y2:GOTO line7
1250  Y5=BY1-XB/(YB*2)
1260  IF Y5<BY1-W(A+1)*YA THEN Y5=Y2: X4=X3
1270  line7:  LINEF X1,BY1,X4,Y5          : REM 7
1280  IF (A=NR) THEN Y4=Y2  : GOTO line8
1290  Y4=(BY1-(W(A+1)*YA))-XB/(YB*2)
1300  IF (Y4>Y2) THEN Y4=Y2
1310  IF (Y4<Y3) THEN Y4=Y3
1320  line8: LINEF X3,Y3,X3,Y4    : REM 8
1330  COLOR 1,1,1,A,2
1340  FILL X2+(X1-X2)/2,BY1-(BY1-Y1)/2,1
1350  NEXT A
1360  COLOR 1,0,1,0,1
1370  RETURN
1380  '-----------------------------------------------
1500  beschriftung:
1510  GOTOXY 0,3: PRINT YM
1520  GOTOXY 0,9: PRINT YM/2
1530  FOR A=NR TO 1 STEP -1
1540  X=(A*XA-XB/2)/8
1550  IF A MOD 2 =0 THEN Y=16 ELSE Y=15
1560  GOTOXY 0,Y :PRINT SPC(X)  W(A)
1570  NEXT A
1580  RETURN
1590  '-----------------------------------------------
2000  diagramm :
2010  LINEF DX1,DY1,DX2,DY1
2020  LINEF DX2,DY1,DX2,DYY
2030  LINEF DX1,DY1,DX1,DYY
2040  LINEF DX1,DY1-YM*YA,DX2,DY1-YM*YA
2050  LINEF DX1,DY1-(YM/2)*YA,DX2,DY1-(YM/2)*YA
2060  FOR A=1 TO NR-1
2070  X1=(A*XA)-XA/4
2080  Y1=DY1-W(A)*YA
2090  X2=((A+1)*XA)-XA/4
2100  Y2=DY1-W(A+1)*YA
2110  LINEF X1,Y1,X2,Y2
2120  NEXT A
2130  COLOR 1,0,1
2140  RETURN
2150  '-----------------------------------------------
3000  kuchengrafik:
3010  CIRCLE CX1,CY1,R1
3020  W1=3600/SU : WE=0
3030  FOR A=1 TO NR
3040  WA=WE
3050  WE=WE+W(A)*W1
3060  COLOR 1,1,1,A,2
3070  PCIRCLE CX1,CY1,R1,WA,WE
```

```
List of \GRAPH.BAS

 3080  NEXT A
 3090  COLOR 1,0,1
 3100  RETURN
 3110  '-------------------------------------------------------------
 3500  kuchenbeschriftung :
 3510  FOR A=1 TO NR
 3520  COLOR 1,1,1,A,2
 3530  CIRCLE 500,A*(340/20)-5,8 : FILL 500,A*340/20-5
 3540  GOTOXY 30,A-1: PRINT W(A)
 3550  NEXT A
 3560  COLOR 1,0,1
 3570  RETURN
 3580  '-------------------------------------------------------------
```



6000 Frankfurt/M.

Müller & Nemecek
Kaiserstraße 44
6000 Frankfurt/M.
Tel. 0 69 - 23 25 44

WAIZENEGGER
Büroeinrichtungen
Kaiserstraße 41
6000 Frankfurt/M.
☎ 0 69 / 23 92 31

6100 Darmstadt-Eberstadt

Heim-Verlag
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
✆ 0 61 51 / 5 53 75

6240 Königstein

KFC COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Telefon 0 61 74 / 30 33

6457 Maintal/Dörnigheim

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

Elphotec Computer Systeme
Ihr Atari Systemhändler mit eigenem Service-Center
Schießgartenstraße 7
6500 Mainz
Telefon 0 61 31 - 23 19 47

6520 Worms

ORION
Computersysteme GmbH
Friedrichstraße 22
6 5 2 0 W O R M S
Tel. 0 62 41 / 67 57 - 67 58

6700 Ludwigshafen

MKV Computermarkt
Rathauscenter
6700 Ludwigshafen
Telefon 06 21 - 52 54 95

6720 Speyer

MKV Computermarkt
Gilgenstraße 4
6720 Speyer
Telefon 0 62 32 - 7 72 16

6750 Kaiserslautern

C.O.S.
COMPUTER ORG. GmbH
Karl-Marx-Straße 8
6750 Kaiserslautern
Telefon (06 31) 6 30 71 - 74

6800 Mannheim

Computer Center
Am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim
Tel. 06 21 / 2 09 83 - 4

Horten
COMPUTERCENTER
N 7
6800 Mannheim
✆ 06 21 - 1 30 91

6900 Heidelberg

Heidelberger Computer-Center
Bahnhofstraße 1
6900 Heidelberg
Telefon 0 62 21 / 2 71 32

JACOM COMPUTERWELT
Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 0 62 21 / 41 05 14 - 550

7000 Stuttgart

BNT Computerfachhandel Seibel & Co. oHG
Der kleine mit der großen Leistung
Markstraße 48 · 7000 Stuttgart 50
Telefon 07 11 / 55 83 83

7022 Echterdingen

Autorisierter ATARI-System-Fachhändler für **520 ST** 130 XE
ATARI
Matrai computer
Michael Matrai
Bernhauser Str 8
7022 L. Echterdingen
☎ (07 11) 79 70 49

7100 Heilbronn

Unser Wissen ist Ihr Vorteil
Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048

Seel's Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 03

8400 Regensburg

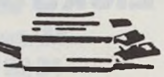
Zimmermann elektroland
8400 Regensburg · Dr.-Gessler-Str. 8 · ☎ 09 41/9 50 85
8390 Passau · Meraner-Str. 5 · ☎ 08 51/5 10 16

8490 Cham

AGP-SHOP®
INH. A. STEUER
Auf der Schanze 4 · 8490 Cham/Opf. · Telefon (09971) 9723



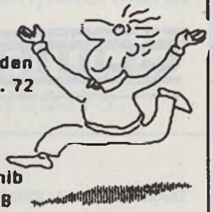
Ihr Fachgeschäft für Microcomputer und Schreibwaren



Zoom – Fotokopieren

8500 Nürnberg

HIB-GMBH Computerladen
Äußere Bayreuther Str. 72
Postfach 21 01 25
8500 Nürnberg 21
Tel: 0911 / 515 939
Telex: 17 - 911 8253 hib
Teletex: 911 8253 HIB



8520 Erlangen

BÜRO 2000 HAAS
Dresdener Str. 5 · Friedrichstr. 9
8520 Erlangen · Telefon 12 01-0

Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76

8590 Marktredwitz

STEINWALD ELEKTRONIK GMBH
Am Steingrund 1 · 8590 Marktredwitz
Telefon 0 92 31/ 6 20 18

8600 Bamberg

BÜROZENTRUM A. + R. KUTZ
Am Kranen 12a · 8600 Bamberg
✆ 09 51/2 78 08 · Telex 662 494

8700 Würzburg

HALLER GMBH
Fachgeschäft für Mikrocomputer
Büttnerstraße 29
8700 Würzburg
Tel. 09 31/1 67 05

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

8752 Mömbris

COMPUTIQUE
DIE EXPERTEN FÜR MICROCOMPTER
Im Kahltal-Zentrum, 8752 Mömbris
Telefon: (060 29) 65 20 oder 14 10
ATARI 520 ST · APRICOT · IBM Komp · Festplatten
Hardware · Software · Schulung

8850 Donauwörth

ORG technik
☎ 09 06 - 60 99
Altes Sträßle 28
8850 Donauwörth
Ihr Büroausstatter

8900 Augsburg

Vertragshändler für IBM Personal-Computer
PROFI COMPUTER STUDIO
KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg
Telefon(08 21)31 53-4 16

8900 Augsburg

Adolf & Schmoll Computer Studio
Hörbrotstr. 6 · 8900 Augsburg
Telefon (08 21) 52 85 33
Wir sind außerdem autorisierte Service-Fachwerkstatt für:
Schneider · commodore · ATARI

8901 Augsburg-Vogelsang

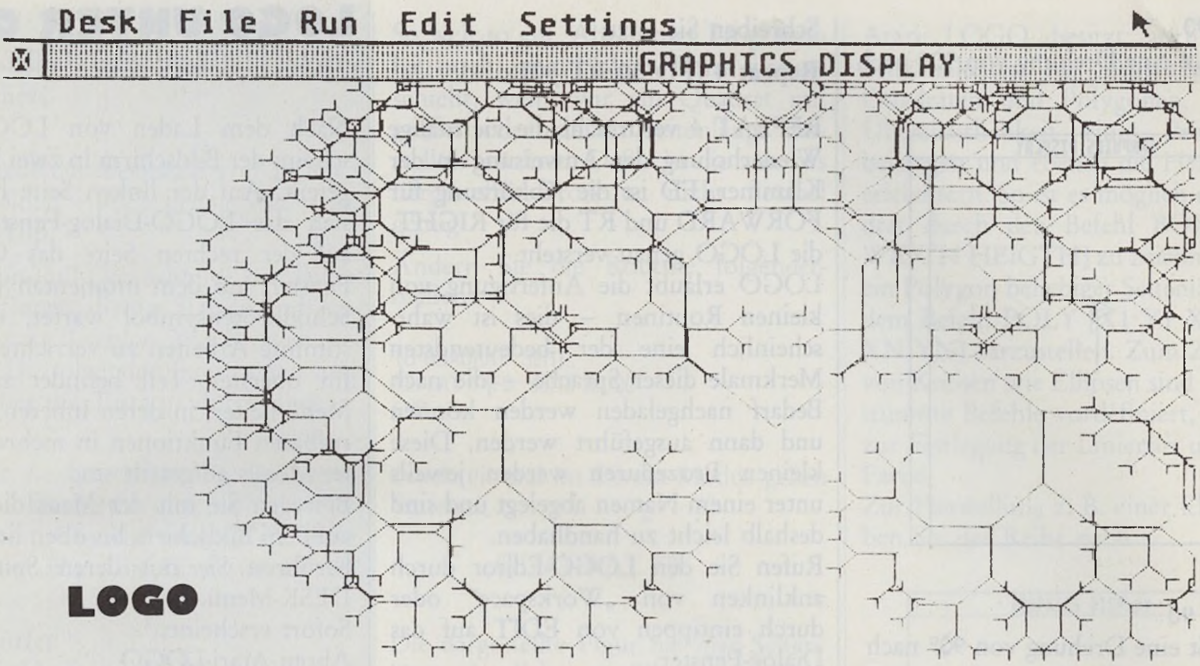
VIDEO + COMPUTER HANDELS GMBH
Steppacher Straße 8 A
8901 Augsburg-Vogelsang
Telefon 08 21 / 48 20 76

8910 Landsberg

Szeredy Elektronik GmbH
Computervertrieb
Max-Von-Eyth-Str. 4
8910 Landsberg · ✆ 0 81 91 / 3 95 00

8940 Memmingen

EDV-Organisation Hard- + Software
Manfred Schweizer KG
Benninger Str. 34, Tel. 0 83 31 / 1 22 20
8940 Memmingen



LOGO

– oder eine Einführung in Turtle-Geometrie

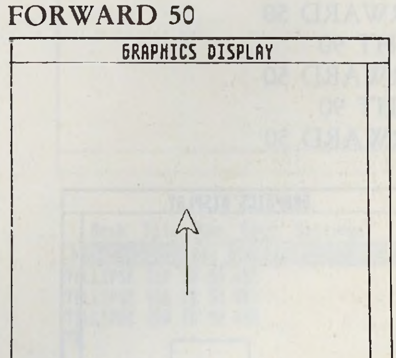
Unbestritten ist BASIC die verbreitetste Sprache für Personal- und Home-Computer. Es liegt vielleicht an der Tatsache, daß die Mehrzahl dieser Computer vom Werk her mit einem BASIC-Interpreter ausgerüstet sind.
Atari hat diese Philosophie geändert. Zu dem Lieferumfang seiner neuen Generation von Computern (ST) gehören neben BASIC auch LOGO von Digital-Research.
Samt Maus liefert Atari zu seiner ST-Computer-Reihe ein anderes Tierechen in Form einer Schildkröte und damit eine leistungsfähige Computersprache, nämlich D.R. LOGO.
LOGO wurde Ende der 60er Jahre von Seymour Papert, Professor in dem Institut of Technology, Massachusetts, entwickelt; ein Labor, welches sich mit der künstlichen Intelligenz beschäftigt.
Papert, ein Mathematik-Professor und ehemaliger SDtudent des schweizerischen Psychologen Jean Piaget (bekannt als Vater der modernen Psychologie), strebte nicht nach einer neuen Sprache, die ein Optimum eines Computers bilden sollte, sondern eher nach einem pädagogischen Werkzeug, daß die intellektuellen Fähigkeiten der Studierenden entwickeln sollte – genauso wie der tatsächliche Lernprozeß komplizierte Vorgänge erleichtert.

So entstand eine Programmiersprache, die in Dialog-Betrieb mit dem Anwender arbeitet und in der Lage ist, schwierige Prozesse anschaulich zu lösen; und daß durch die Ausführung von kurzen Befehlen, die die Form von einfachen und bekannten Begriffen haben. Bei dem damaligen Stand der Technik war es unmöglich solche Lösungen auf einem Sichtgerät darzustellen. Papert schuf eine mechanische „Schildkröte", an dessen Unterseite ein Stift befestigt war. Beim Fahren auf einer Papierunterlage hinterließ sie Linien. Heute hat sich diese motorisierte „Schildkröte" in ein stilisiertes Dreieck verwandelt, die auf dem Bildschirm wunderschöne Figuren zeichnen kann.

Wenn der griechische Mathematiker Phythagoras im Zeitalter der Computer gelebt hätte, wäre ihm LOGO mit großer Wahrscheinlichkeit als Sprache zur Lösung seiner wichtigen mathematischen und geometrischen Probleme dienlich gewesen. Mit einfachen Befehlen ist zu beweisen, daß die kürzeste Entfernung zwischen zwei Punkten eine Strecke ist. Dazu braucht man nur FORWARD zu schreiben, zusätzlich die gewünschten Schritte einzugeben und schon bewegt sich die Schildkröte.

Tippen Sie:

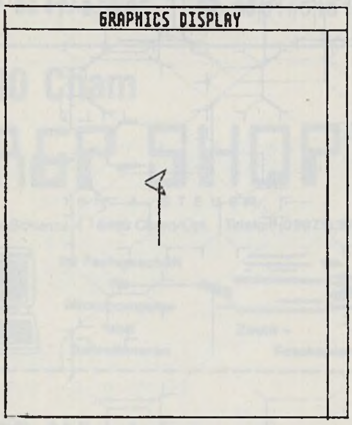
FORWARD 50



Diese Eingabe veranlaßt unser Symbol sich 50 Schritte in die Richtung, in die die Spitze zeigt, zu bewegen. Dabei wird eine Linie auf dem Bildschirm gezogen. Mit BACK, wieder gefolgt von einer numerischen Angabe, rückt die Schildkröte zurück.

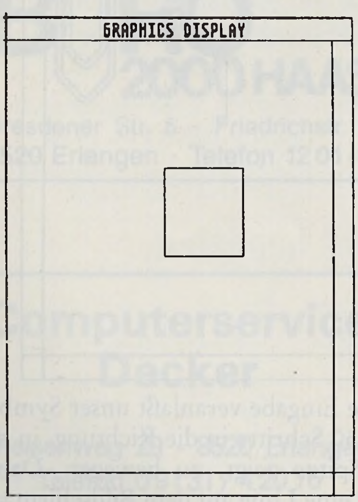
BACK 50
Setze die Schildkröte an ihren ursprünglichen Platz; die vorher gezeichnete Linie bleibt dabei erhalten.
Genauso kann man die Laufrichtung ändern

LEFT 90
Veranlaßt eine Drehung von 90° nach links.



RIGHT 90
Veranlaßt eine Drehung von 90° nach rechts.

Mit diesen vier primitiven, aber leistungsfähigen Befehlen können wir anfangen zu zeichnen.
Geben Sie ein:
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50



Auf dem Output-Fenster wird schrittweise ein Quadrat gezeichnet. Man sollte aber nicht verschweigen, daß für so wenig ein solcher Aufwand betrieben wird. Aber es geht auch anders – viel leichter und schneller.
LOGO besitzt einen Befehl, der REPEAT heißt. Damit sieht unsere Prozedur, ein Quadrat zu zeichnen, etwas anders aus.

Schreiben Sie jetzt:
Repeat 4 [FD 50 RT 90]

REPEAT 4 verursacht die viermalige Wiederholung der Anweisung in der Klammer. FD ist die Abkürzung für FORWARD und RT die für RIGHT, die LOGO genau versteht.
LOGO erlaubt die Anfertigung von kleinen Routinen – dies ist wahrscheinlich eine der bedeutendsten Merkmale dieser Sprache – die nach Bedarf nachgeladen werden können und dann ausgeführt werden. Diese kleinen Prozeduren werden jeweils unter einem Namen abgelegt und sind deshalb leicht zu handhaben.
Rufen Sie den LOGO-Editor durch anklinken von „Workspace" oder durch eintippen von EDIT auf das Dialog-Fenster.
Jetzt befinden Sie sich im Editor-Modus, und das Schreiben eines Programms ähnelt sehr stark dem Schreiben eines gewöhnlichen Texts. Sie müssen nur auf bestimmte, von LOGO verlangte, Syntaxregeln achten. Wie könnte dies anders ein? Es handelt sich schließlich um eine Sprache.
Schreiben Sie jetzt:

```
TO QUADRAT
    REPEAT 4 [FD 50 RT 90]
```

In direktem Modus geben Sie
QUADRAT
ein und die Schildkröte beginnt zu zeichnen. Kurz darauf steht unser wohlbekanntes Quadrat auf dem Grafik-Display.
Das kann man jetzt abspeichern und bei Bedarf immer abrufen. Auf diese Art und Weise wird der Computer eine riesige Menge an Funktionen beherrschen, die man später zur Lösung anderer Probleme anwenden kann.

LOGO UNTER GEM

Nach dem Laden von LOGO erscheint der Bildschirm in zwei Hälften geteilt. Auf der linken Seite befindet sich das LOGO-Dialog-Fenster und auf der rechten Seite das Output-Fenster, auf dem momentan nur das Schildkrötensymbol wartet, um bestimmte Arbeiten zu verrichten.
Im obersten Teil befindet sich die Menü-Leiter, in deren Inneren die abrufbaren Funktionen in mehrere Unter-Menüs aufgeteilt sind.
Bewegen Sie mit der Maus die Pfeile auf dem Bildschirm bis oben links und berühren Sie mit deren Spitze das DESK-Menü.
Sofort erscheint:
About-Atari-LOGO
VT 52 Emulator
Kontrollfeld RS 232 Einstellung
Druckeranpassung
Das DESK-Menü ist schon durch das Atari-Handbuch bekannt und ausführlich erklärt.
Das zweite nach rechts stehende Pull-Down-Menü trägt den Namen „File" und besteht aus:

Load
die auf Disketten gespeicherten Programme erscheinen in einer „Dialog-Box", die mit Hilfe der Maus ausgewählt und in den Arbeitsspeicher geladen werden.

Save us
ermöglicht das Kopieren eines in der Diskette schon vorhandenen Programms auf der selben Diskette. Vorsicht! Nicht zu verwechseln mit dem BASIC-Kommando „Rename".

Save
ein fertig editiertes Programm läßt sich damit auf Diskette speichern .

Delete
gespeicherte Files werden dadurch gelöscht.

Load-Pic
ein auf Diskette gespeichertes Grafik-Bild wird in den Grafik-Puffer geladen. Die Größe des Grafik-Display entspricht der Originalgröße.

Save-Pic
eine erstellte Grafik wird auf Diskette gespeichert.

Quit
verursacht das Verlassen von LOGO und einen Rücksprung in das DESK-Top.

Das Run-Pull-Down-Menü beinhaltet die Optionen, die das Starten und Anhalten von Befehlen und Prozeduren regeln. Die folgenden Funktionen sind auch über die Tastatur durchführbar:

Stop
beendet die Durchführung eines Programms. Durch das gleichzeitige Betätigen von Control und C ist diese Funktion über die Tastatur zu erreichen.

Run-Buffer
beim Anklinken dieser Funktion wird der letzte direkte Befehl oder das Programm wieder ausgeführt. Ein Clear-Text oder Clear-Screen löscht den Run-Buffer.

Pause
hält den Programmablauf an und läßt eine Eingabe über die Tastatur zu. Durch das Auswählen von Continue aus dem gleichen Pull-Down-Menü, wird das Programm fortgesetzt.

Continue
s. o. Continue erreicht man auch mit Hilfe der Abkürzung CO und Return.

Das Edit-Menü ist das umfangreichste und wichtigste Menü von allen. Dadurch hat man Zugang zu einem sehr leistungsfähigen Editor.

Work-Space
Öffnet das Editor-Fenster zum Editieren von Programmen.

File
Lädt eine schon vorhandene Prozedur zur Weiterverarbeitung oder Änderung, in den Speicher.

Save Edit ¯C
Speichert ein editiertes Programm auf Diskette.

Abandon ¯C
Verlassen des Editors und Zurückkehr in das Dialog-Fenster. Dadurch geht der Inhalt des Editor-Fenster verloren.

So weit so gut. Aber unser Programm hat noch einen Haken. Denn was geschieht, wenn wir ein Quadrat mit großer oder kleiner Seitenlänge darstellen möchten. Dafür ist die von uns hergestellte Prozedur nicht flexibel genug.

Ändern Sie die Routine folgendermaßen:

```
TO QUADRAT :SEITE
REPEAT 4 [FD :SEITE RT 90
END
```

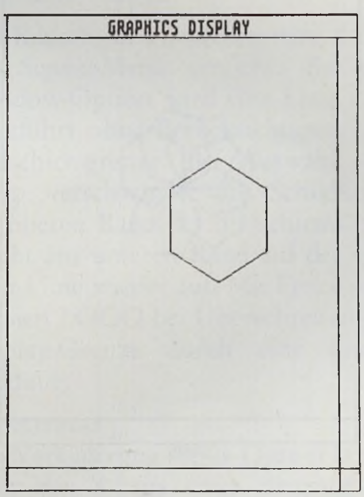
Noch einmal im Direkt-Modus geben Sie ein:

QUADRAT 70

Die dargestellte Figur hat ihre Seitenlänge deutlich vergrößert. Auf diese Art und Weise ist es uns möglich das schon fertige Programm mit geringen Änderungen so zu gestalten, daß es etliche Polygonen zeichnen kann.
Ändern Sie die Routine folgendermaßen:

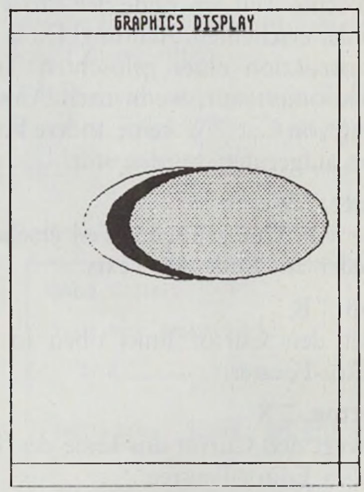
```
TO QUADRAT :SEITE :WINKEL :WIE
REPEAT :WIE [FD :SEITE RT :WINKEL]
```

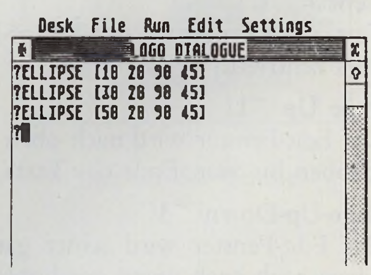
Rufen Sie die Routine so auf:
QUADRAT 50 72 5
Jetzt haben wir eine geometrische Figur mit fünf Seiten. Bei
QUADRAT 50 60 6
eine mit sechs Seiten usw.



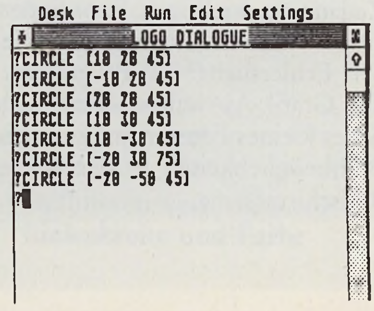
Wir haben die QUADRAT Prozedur derart verändert, daß der Name in keiner Weise mehr stimmt. Der passende Name wäre jetzt Polygon.

Atari LOGO besitzt auch andere Möglichkeiten zur Darstellung von Quadraten und Polygonen, die zur Übersichtlichkeit eines Programms beitragen und ebenso die Herstellung erleichtern. So ist es möglich ein Quadrat durch den Befehl BOX [X Y WIDTH HEIGTH] zu zeichnen, oder ein Polygon beliebiger Seitenlänge mit dem Befehl POLY [X1 Y1 X2 Y2 ... XN YN] darzustellen. Zum Zeichnen von Kreisen wie Ellipsen sind auch bestimmte Befehle vordefiniert, genauso zur Festlegung der Linienart und ihrer Farbe.
Zur Herstellung z. B. einer Ellipse geben Sie der Reihe nach





Wobei die Angabe der Anfangskoordinate, sowie die Höhe und Breite der Winkel dargestellt werden soll.
Zum Zeichnen von Kreisen geben Sie der Reihe nach



Mark ¯S
Ein bestimmter Bereich oder eine ganze Prozedur wird zwischengespeichert. Mit Copy ¯T läßt sich dieser Bereich an das Ende des Texts kopieren. Damit erhält der Programmierer ein leistungsfähiges Werkzeug um Teile eines Programms anzuhängen.

Copy ¯T
Gewählter Bereich wird am Ende des Texts abgetrennt.

CuT ¯W
Falls ein Teil des Programms gelöscht werden muß, läßt es sich auf diese Art und Weise realisieren. Bei der Rückgängigmachung des Vorhabens wird durch Auswählen von Taste ¯Y der gelöschte Teil am Ende der Prozedur wieder erscheinen. Achtung! Diese Rekonstruktion eines gelöschten Texts funktioniert nur, wenn nach Anwendung von Cut ¯W keine andere Funktion aufgerufen werden soll.

Taste ¯Y
Der von Cut gelöschte Teil erscheint wieder am Ende des Texts.

Top ¯R
setzt den Cursor links oben in das Editor-Fenster.

Botton ¯X
bewegt den Cursor ans Ende des Textes im Editor-Fenster.

Center ¯L
Text wird innerhalb des Editor-Fensters zentriert.

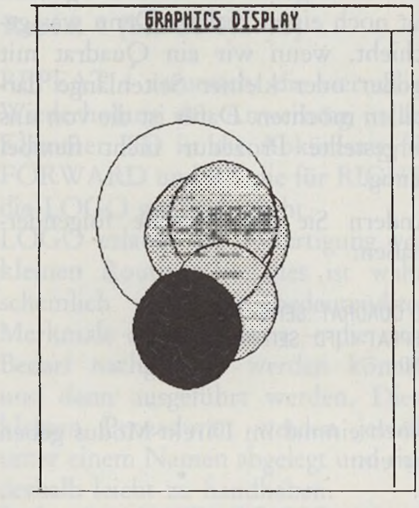
Page Up ¯U
Das Edit-Fenster wird nach oben verschoben bis zum Ende des Texts.

Page-Up-Down ¯V
Das Edit-Fenster wird seiner ganzen Länge nach nach unten geschoben.

< = Line ¯A
Cursor rückt bis zur Beginn der Linie.

= > Line ¯E
Cursor rückt bis zum Linienende.

Über das Menü „Setting“ erhält man Zugang zu wichtigen Funktionen für grafische Vorbereitungen, sowie bei der Fehlersuche im Programm. Bei den Grafik-Auswahlen öffnet sich ein neues kleines Fenster mit zusätzlichen Wahlmöglichkeiten in der Mitte der Bildschirmfläche.

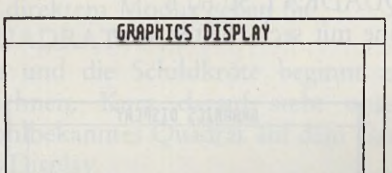


Wie schon erwähnt könnte eine schon hergestellte Prozedur innerhalb einer anderen aufgerufen und ausgeführt werden. Wir definieren ein Modul unter dem Namen Hexagon, das ein Polynom mit sechs Kanten zeichnet.

```
TO HEXAGON :SEITE
REPEAT 6 [FD :SEITE RT 60]
END
```

Jetzt schreiben wir ein Programm, das die Prozedur Hexagon aufruft.

```
TO SUPERHEX :SEITE
REPEAT 7 [HEXAGON:SEITE LEFT 45]
END
```

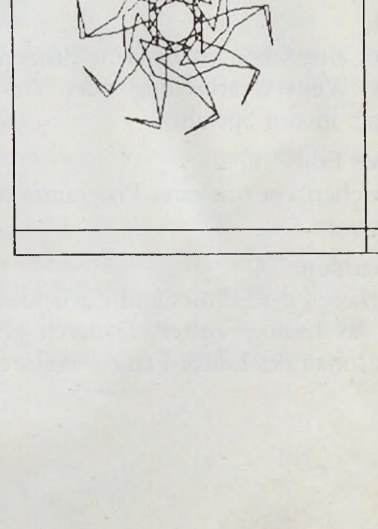


Auf diese Art und Weise können wir ein Programm erweitern. Eine solche Form der Selbstaufrufung nennt man REKURSION und sie ist eine der wichtigsten Eigenschaften von LOGO. Man muß allerdings sagen, daß unsere jetzige Form von Rekursion nicht unbedingt die beste ist. Vor allem, weil die Anzahl der Selbstaufrufe nicht ohne größere Eingriffe in das Programm selbst zu steuern ist. Betrachten wir das nächste Beispiel:

```
TO SPIRAL :SEITE :WINKEL :INS :INW
FD :SEITE
RT :WINKEL
SPIRAL :SEITE ★ :INS :WINKEL ★ :INW :INS :INW
END
```

Es wird eine Prozedur definiert, die vier Eingaben zuläßt: Seite, Winkel, Inkrementseite, sowie Inkrementwinkel. Dann wird eine Strecke gemäß unserer Eingabe in einem bestimmten Winkel gezeichnet. Jetzt ruft sich die Prozedur selbsdt auf, nur daß dabei die Strecke, sowie der dazugehörige Winkel inkrementiert wird. Geben Sie jetzt in direktem Modus ein:

```
?CS
?SPIRAL 30 12 0 80
```

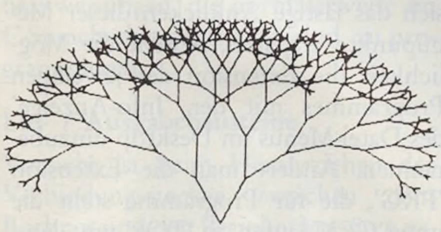


Wie Sie sehen hört es nie auf. Erweitern wir unser Programm SPIRAL folgendermaßen:

```
TO SPIRAL :SEITE :WINKEL :INS :INW
IF :WINKEL = 360 [STOP]
FD :SEITE
RT :WINKEL
SPIRAL :SEITE ★ :INS :WINKEL ★ :INW :INS :INW
END
```

Wie Sie sehen, arbeitet die Anweisung IF genauso wie bei anderen Sprachen und gibt die Möglichkeit zur Steuerung einer Schleife. Damit ist dieser Teil unserer Reihe über das Programmieren von LOGO beendet. Das nächste Mal werden wir uns mit STRING-Verwaltung, sowie LISTEN und weiteren Anweisungen beschäftigen.
Zuletzt ein Leckerbissen von LOGO:

```
TO B1 :N
IF :N = 0 [STOP]
LT 25 FD 5 ★ :N
B1 :N – 1
BACK 5 ★ :N RT 25
RT 25 FD 5★ :N
B1 :N – 1
BACK 5 ★ :N LT 25
END
```



Fill
Durch Auswahl der Frage „True" ist es möglich die Art der Muster zu definieren. Bei Farbmonitoren auch die Farbe.

Zoom
bewirkt eine Vergrößerung.

Line
gibt die Möglichkeit Typen und Linienbreiten auszuwählen; sowie bei der oben genannten Fill-Option besteht auch hier die Möglichkeit zur Wahl der Linienfarbe. Vorausgesetzt man besitzt einen Farbmonitor.

Auch bei dem Turtle-Menü erscheint eine Dialog-Box mit verschiedenen Auswahlmöglichkeiten zur Steuerung des Schildkrötensymbols.

Position

Heading
Der Standpunkt der Schildkröte wird festgesetzt. Die Spitze zeigt die Fahrtrichtung an.

X:_ Y:_
Eine neue Position der Schildkröte wird festgesetzt, in Bezug auf X und/ oder Y Koordinate.

Turtle State
man kann damit zwischen einer Darstellung mit oder ohne dreieckigem Symbol wählen.

Pen State
Die Schildkröte stellt in LOGO den fahrenden Stift dar. Man kann durch Pen-State den benannten Stift hoch und runter fahren.

Die Bildschirmverwaltung wird durch das Screen-Menü erreicht. Bei der Window-Option wird eine Linie weitergeführt ohne Berücksichtigung der Bildschirmgrenze. Bei Auswahl von Wrap verschwindet die Schildkröte am oberen Rand des Bildschirms und taucht am unteren Rand auf der gleichen Linie wieder auf. Mit Fence meldet sich LOGO bei Überschreiten der Display-Grenze durch eine Error-Meldung.

Aspectratio
Die Veränderung dieser Option beeinflußt die Y-Lage einer dargestellten Grafik.

Pyn
Damit kann man wahlweise die X- oder Y-Lage verändern.

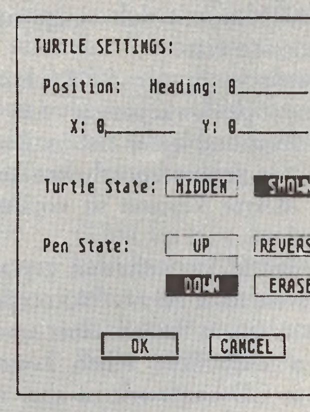
Mit Hilfe von Watch und Trace wird in ein dafür extra geöffnetes Fenster, Debug-Info, der Programmablauf schrittweise gezeigt.

Buffer-Grph
Für die Grafik-Fenster reserviert LOGO normalerweise einen bestimmten Bereich. So kann das Grafik-Display durch ein anderes Fenster verdeckt werden, ohne daß dadurch die dargestellte Grafik verlorengeht. Dieser Puffer kann beliebig oft ein- und ausgeschaltet werden.

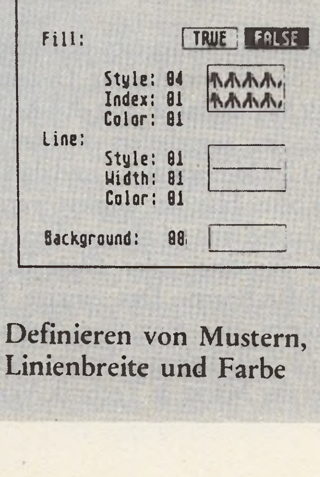
Close-Debug
beendet die Trace-Funktion und schließt das Debug-Fenster.

Close-Edit
beendet die Editor-Funktion, läßt aber das Edit-Fenster offen.

Alle Fehlermeldungen werden auch in Form einer Dialog-Box mitten auf dem Bildschirm gezeigt.



Definieren der Turtle



Definieren von Mustern, Linienbreite und Farbe

NEWS

Zuerst einmal ein Wort in eigener Sache. Wir – die ST-Redaktion – versuchen in diesen News soviel wie möglich neue Informationen für Sie – die Leser – zu sammeln. Dabei sind wir aber auch redlich bemüht, über keine Gerüchte, sondern ausschließlich über Fakten und fest geplante Projekte zu berichten. Um die nötigen Informationen zu bekommen, versuchen wir einen engen Kontakt zu Software-Herstellern und der Firma ATARI selbst zu unterhalten. Dies unterscheidet uns von einigen anderen Computerzeitschriften, die nur auf Neuigkeiten aus sind, egal, ob sie der Realität entsprechen oder nicht.

Der ATARI 520ST ist zum **Computer des Jahres 1985** in der Sparte Personalcomputer gewählt worden. Dieser Titel wird jedes Jahr von internationalen Fachjournalisten aus sieben Ländern vergeben.

Nach einem längeren Telefongespräch mit ATARI stellten sich einige neue Informationen ein:

Die **Festplatte** zu den ST-Rechnern wird eine Speicherkapazität von 20 MByte haben und leider erst im Laufe des nächsten Jahres zu erhalten sein. Eine 10 MByte Version ist vorläufig nicht geplant.

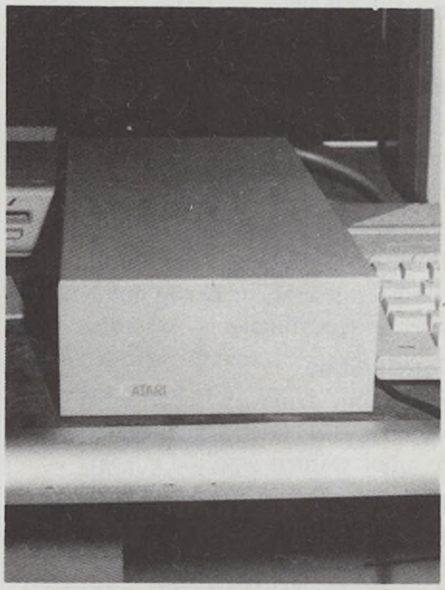
Das **CD-ROM** ist weiterhin geplant. Es existieren auch bereits Prototypen, wie man in einem Bericht einer amerikanischen Zeitschrift sehen konnte. Mit einem CD-Laufwerk ist aber nur das Lesen von Daten möglich. So werden später wohl nur mehrbändige Lexika o. ä. auf den kleinen silbernen Scheiben erhältlich sein.

Weiterhin gibt es zu berichten, daß das **TOS im ROM** bis zum Jahresende wahrscheinlich nicht mehr ausgeliefert wird.

Auch software-mäßig gibt es allerhand Neues. So ist seit Anfang Dezember ein **CP/M-Emulator** von ATARI an die Händler ausgeliefert worden. Er hat das CP/M-Format 2.2 und darf **kostenlos** beim Händler kopiert werden. Damit laufen nun so gängige Programme wie Wordstar oder dBase III auf den ST-Rechnern. Das einzige Problem bleibt, wie man diese Software auf 3 1/2-Zoll-Disketten bringt. Abhilfe bietet der Markt & Technik Verlag. Bei ihm ist diverse Software auf CP/M 2.2 Format für die ATARI-Laufwerke zu erhalten.

Desweiteren testet ATARI derzeit ein **Fakturierungs-** und **Lagerverwaltungsprogramm**, das von der Münchner Softwarefirma SM hergestellt wurde. Es soll dann rechtzeitig zu Weihnachten auf den Markt kommen und direkt bei ATARI zu beziehen sein.

Zuletzt wurde uns noch versichert, daß an den bisher bei der Lieferung der 520ST-Rechner fehlenden Programmen **GEM-Write** und **GEM-Paint** fieberhaft gearbeitet würde und sie nach ihrer Fertigstellung sobald wie möglich beim Händler nachzubeziehen seien.



Prototyp der 20 MByte Festplatte

Informationen

Verschiedene TOS-Versionen

Zur Zeit sind drei TOS-Versionen im Umlauf, die allerdings unterschiedliche Fehler aufweisen. So kann man zum Beispiel am seriellen Bus (RS 232) keinen Drucker anschließen. Der Rechner erwartet, über diese Schnittstelle nach jedem gesendeten Zeichen eine Empfangsbestätigung von der Peripherie. Diese Bestätigung (Handshake) wird von einem Drucker nicht geschickt. Ein Drucker liest eine größere Menge von Zeichen bis sein Puffer voll ist. Bei einem Modem funktionierte diese Schnittstelle einwandfrei, da Modems das benötigte Handshake-Signal geben.

Die Druckqualität bei der implementierten Hardcopyroutine ('Alternate + Help'), läßt sich nicht ändern. Der Ausdruck ist gleich, unabhängig von der Einstellung 'Test' oder 'Qualität'. Ebenso findet keine Überprüfung der Uhrzeit im Kontrollfeld statt.

Bildschirminvertierung

Durch Nullstellen der RGB-Regler im Kontrollfeld erfolgt beim monochromen Monitor eine komplette Invertierung des Bildschirms. Die Schrift erscheint in Weiß auf schwarzem Grund.

TOS-Kopie

Die TOS-System Diskette kann nur auf einseitig formatierten Disketten kopiert werden. Das Boot-Programm ist nicht für doppelseitiges Format ausgelegt. Da man mit der SF 314 Floppy auch einseitiges Format beschreiben kann, ist es natürlich auch möglich sich von der TOS-Diskette Backups zu machen.

Programme ohne GEM

Wenn man schon etwas mehr mit Anwenderprogrammen für die ST-Serie zu tun hatte, ist einem bestimmt schon der Punkt 'Anwendung anmelden' im Option-Menü des Desktops begegnet. Mit ihm ist es möglich zwischen 'GEM', 'TOS' oder 'TOS übernimmt Parameter' umzuschalten. Will man sich das lästige Anklicken dieser Menüpunkte ersparen, besteht die Möglichkeit die Extension des jeweiligen Programmes mit der 'Info-Anzeige' des Datei-Menüs im Desktop umzubenennen. Ändert man die Extension 'PRG', die für Programme steht die unter GEM laufen, in 'TOS' um, dann wird bei jedem Laden des jeweiligen Programmes automatisch auf das TOS umgeschaltet. Wenn man nun will, daß das TOS auch noch Parameter übernimmt, wie es zum Beispiel bei den meisten Hilfsprogrammen der Utility-Diskette des Entwicklungspaketes der Fall ist, dann gibt man die Extension 'TTP' an und es meldet sich beim Laden das entsprechende Fenster auf dem Bildschirm.

Man spart sich durch diese Extensionangabe das erneute Auswählen vor jedem Laden.

Tastaturbelegung

Die Tastatur der Atari ST sendet keine ASCII-Zeichen. Der eventuelle Anschluß einer anderen Tastatur ist somit nicht problemlos möglich. Die gesendeten Tastaturdaten werden erst im TOS ausgewertet und in ASCII-Zeichen umgewandelt. Hier besteht die Möglichkeit die Tastaturbelegung umzudefinieren.

On Line

Man kann mit dem im Desktop implementierten VT-52 Emulator und einem handelsüblichen Modem bzw. Akustikkoppler sofort ON LINE gehen. Als Verbindung wird ein normales RS 232C-Schnittstellenkabel mit einer 1:1 Belegung benötigt. Zuvor sollte man für die Kommunikation mit normalen Mailboxen die Baud-Rate der Schnittstelle auf 300 Baud einstellen.

Entwicklungspaket

Am Entwicklungspaket zum Atari ST, in dem sich unter anderem ein 63 000 Assembler und ein C-Compiler befindet, wurden schon des öfteren Änderungen vorgenommen. In der neuesten Version wurde der Editor von Metacomco durch den ST Textmanager ersetzt. Als Zusatz zum C-Compiler gibt es jetzt auch die von so manchem Programmierer lang ersehnte Standardbibliothek 'stdio.h'. Diese wird benötigt um Eingabe- und Ausgabeanweisungen, die normalerweise im C-Sprachschatz enthalten sind, zu programmieren.

Ein- / Ausgabebausteine

Vorsicht ist beim Herausziehen der Verbindungsstecker zwischen dem Rechner und den Peripheriegeräten geboten. Manche Ports stehen auch beim abgeschalteten Gerät unter Spannung. Dies betrifft die +/- 12 Volt Betriebsspannung, die bei der RS-232 benötigt wird. Deshalb sollte man bei der Verkabelung und zur eventuellen Kabelentwirrung, den Rechner vom Netz trennen oder die Stromzufuhr zu diesen Geräten unterbrechen. Die Portbausteine der Geräte werden Ihnen dankbar sein.

Bücher

GEM für den Atari 520 ST

Am Anfang war das Buch, und dann kam der Computer. Kurz nachdem Atari seine neue ST-Generation von Computern auf der Hannover-Messe vorgestellt hatte, brachte der Verlag Markt & Technik zwei Bücher über diese Rechnerserie heraus. Eines davon erschien unter dem obengenannten Titel. Zu diesem Zeitpunkt hatte schon ein anderer Verlag ein Buch über den ST veröffentlicht. Aus verschiedenen Gründen werden wir auf dieses Buch jedoch nicht weiter eingehen.

Offensichtlich läßt sich darüber streiten, ob es angemessen ist, ein Buch über einen Computer zu veröffentlichen, ohne daß dieser vorher erschienen ist. In diesem Falle müssen wir es als positiv bewerten, jedenfalls zu einem Zeitpunkt, wo so wenig Konkretes über diesen Computer wie das dazugehörige GEM bekannt war und die breite Öffentlichkeit so wenig davon wußte.

Eines der stärksten Kapitel dieses Buches ist die Beschreibung des GEM-Schreibtisches. Es wird in aller Ausführlichkeit (besser als in dem beiliegenden Handbuch, welches von Atari mitgeliefert wird) die Bedienung mit der Maus von Fenstern, Pull-Down-Menüs, Icons usw. beschrieben. Genauso werden die Handhabung von Dateien und die Möglichkeiten, die dadurch entstehen, erklärt.

Ein anderes Kapitel beschäftigt sich mit der Zusammenarbeit zwischen GEM und Anwender-Programmen. Am Beispiel von LOGO und BASIC (zwei Programmiersprachen, die zum Lieferumfang gehören) wird dem Leser versucht der Gewinn an Leistung, sowie Bedienungsfreundlichkeit, die so ein Programm durch GEM erreicht, deutlich zu machen. Seltsamerweise ist der LOGO gewidmete Teil um einiges besser und umfangreicher beschrieben als der BASIC-Teil. Das lag wohl daran, daß zu diesem Zeitpunkt Atari nicht in der Lage war eine lauffähige Version von BASIC zu liefern.

Das Kapitel über GEM ist wahrscheinlich für den potentiellen Käufer eines der wichtigsten, schließlich ist das GEM ein bei den Atari ST Computern entscheidendes Kaufargument. Das Buch geht zwar nicht sehr ins Detail, aber übermittelt dennoch einen ernsthaften Überblick über eine faszinierende und extrem komplizierte Software Schnittstelle. Es wird in dem Buch auch die allgemeine Unterteilung in VDI und AES ausgeführt. Später werden die wichtigsten Routinen genannt und kurz erklärt.

Zuletzt kommt das schwächste Kapitel überhaupt, über die Hardware des 520 ST. Diejenigen, die über Arbeitsweisen des 68 000er Prozessors etwas erfahren möchten, ist mit einem spezialisiertem Buch eher gedient, als mit diesem Bericht. Das Buch ist (oder war?) eine exzellente Anleitung vor einer Kaufentscheidung. Die Frage ist, ob sich jetzt, nachdem soviele Zeitschriften über das Thema ausführlich berichtet haben, der Kauf eines nicht gerade billigen (52,– DM) Buchs noch rentiert. Aber wie man sagt, bleibt die endgültige Entscheidung bei... dem Leser.

Titel:	GEM für den Atari 520 ST
Autoren:	Josef Steiner/Gerhard Steiner
Verlag:	Markt & Technik 1985 8013 Haan bei München ISBN 3-89090-173-5

Atari ST Intern

Wer schon in Besitz eines anderen Computers ist oder sich allgemein für Computer interessiert hat, wird Data-Becker ein Begriff sein. Die Serie Intern ist schon seit dem Volks-Computer VC 20 und dem Commodore 64 bekannt. Damit hat Data-Becker eine Tradition geschaffen.

Rund 470 Seiten umfaßt dieses neue Werk, und dies sind keine leeren Seiten. Dem Leser kommt eine grandiose Masse von Informationen entgegen, die ohne die nötigen Vorkenntnisse nicht leicht zu verstehen sind.

Anders als im Inhaltsverzeichnis, bei dem das Buch in IC's, Schnittstellen und Betriebssystem eingeteilt wird, ist das Buch unserer Ansicht nach in drei

große Themenkreise zu unterteilen: die Hardware des ST, das Betriebssystem und das BIOS-Listing.
Die Beschreibung der Hardware ist eines der Hauptthemen und von dem Leser nicht leicht zu verstehen. Ein Minimum an Vorkenntnissen in Elektronik sowie Grundkenntnissen über TTL-Technik werden vorausgesetzt, und das nicht ganz zu Unrecht, insofern, daß die neue ST-Generation nicht mehr unter die Kategorie von Home-Computern fällt.
Die Beschreibung aller IC's überfüllt die meisten Seiten des ersten Teils des Hardware Berichts. Der andere Teil beschäftigte sich mit den zahlreichen Schnittstellen, in dem jede einzeln beschrieben wird. Teilweise in einem neuen Stil, der uns eine unbekannte Seite von Data-Becker enthüllt. Gemeint ist hier die nicht ohne Humor beschriebene Funktion des ICs, das das Überwachen der Tastatur übernimmt.

Das zweite und längste Thema ist dem Betriebssystem des Atari ST gewidmet. Nach einer allgemeinen Erklärung über die Bezeichnung sowie die Ähnlichkeit (oder nicht) des ST Betriebssystems mit CP/M oder MS-DOS werden die Funktionen gründlich beschrieben und Beispiele für die Funktionen gezeigt. Gerade hier zeigt dieses Buch seine Stärke. Für jeden, der sich mti dem Betriebssystem auseinandersetzen möchte, ist dieses Kapitel eine obligatorische Lektüre, die man mehrmals wiederholen muß. Innerhalb des genannten Themas wird ausführlich eine Anzahl von Grafik-Routinen, die in das GEM-DOS implementiert sind, beschrieben. Die Routinen dienen der Realisierung von Grafiken auf Maschinensprachebene. Die Nutzung solcher Prozeduren werden durch Beispiele abgeschlossen.
Das BIOS-Listing umfaßt die letzten 150 Seiten und ist vor allem für Software Entwickler von großer Bedeutung. Gerade jetzt, wo das Betriebssystem von der Diskette nachgeladen werden muß, sind Änderungen an diesem System ohne weiteres denkbar. Insofern ist dieses gutdokumentierte Listing für den Programmierer ein unentbehrliches Werkzeug.
Das Intern für den ATARI ST ist ein Muß für jeden Programmierer, der ernsthaft mit seinem Computer arbeiten will und nicht nur fertige Programme bedienen möchte. Es mag wohl sein, daß sich in der Zukunft einiges an den Computern ändern wird, aber man lebt heute und der Computer steht schon auf dem Tisch.

Titel: Atari ST Intern
Autoren: Brückmann/Englisch/Gerits
Verlag: Data-Becker
Merowingerstr. 30
4000 Düsseldorf
ISBN 3-89011-119-X

Leserzuschriften BASIC

Falls Sie Schwierigkeiten mit Hardwareangelegenheiten haben, sei es die Speichererweiterung auf 1 Megabyte, der Anschluß eines anderen Diskettenlaufwerkes oder sonstige technische Probleme, so wenden Sie sich bitte an die ST-Redaktion. Wir sind bemüht, Ihnen bei derartigen Problemen zu helfen.
Andererseits freuen wir uns über Zuschriften betreffs Hardware- oder Softwareerfahrungen.

Manuskripte

Gerne veröffentlichen wir Ihre Beiträge. Dabei sind Programmlistings, Bauanleitungen, Softwaretests oder sonstige Berichte gleichermaßen willkommen. Bei Veröffentlichung erhalten Sie selbstverständlich ein angemessenes Honorar.

CP/M

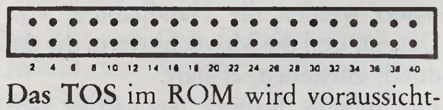
In Zukunft ist eine CP/M-Rubrik geplant, in der über bereits existierende Programme und deren Anpassung an den ATARI ST berichtet wird. Haben Sie eigene Erfahrungen gemacht, so lassen Sie es uns bitte wissen.

ST-Redaktion
Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13

In letzter Minute!

Die Durchnummerierung der Anschlußpins der Erweiterungs-Cartridge (ROM-Moduls) im deutschen Handbuch ist **falsch.** Eine nach diesem Handbuch erstellte Modul-Platine würde einen Kurzschluß der Datenleitung 14 mit +5 Volt erzeugen. Durch diesen schwerwiegenden Fehler könnten einige Chips des Rechners zerstört werden. Die **richtige** Pinnummerierung entnehmen Sie bitte folgender Skizze. Die Signalbezeichnungen der Pins bleiben unverändert.

Cartridge



Das TOS im ROM wird voraussichtlich im ersten Quartal 1986 erscheinen und DM 99,– kosten.

Der **CP/M Emulator** wird doch erst im Januar fertig. Er kann dann bei jedem ATARI Vertragshändler kostenlos auf die eigene Diskette kopiert werden. Dies gilt für **alle** ATARI ST Besitzer.

Vorschau

Die Ausgabe Februar bringt unter anderem:

- Einstellarbeiten am ATARI Monitor (SM 124)
- Softwaretests von:
  SEKA-Assembler von KUMA
  ST-Pascal von CCD
- Eine Einführung in GEM
- Pascal Kurs

Die nächste Ausgabe erscheint Ende Januar.

Für Ihren

ATARI 520/260

VIP Professional

Datenbank/Kalkulation/Grafik

Preis: 748,— DM*

GEM-PASCAL

Preis: 248,— DM*

Bei Ihrem Händler oder
Lieferung frei Haus durch uns
*** unverbindlich empfohlener**
Verkaufspreis inkl. MwSt.

CTK

Computer-Technik Kiechbusch
Am Seeufer 22 5412 Ransbach
Telefon 02623-1618

Qualitätsbücher aus dem *Heim*-Verlag



Buch: 49,- DM
Progr.-Diskette: 58,- DM
Erscheint: Januar 1986

Das Standardwerk für alle ATARI ST-Besitzer. Auf über 300 Seiten eine klare und verständliche Einführung in die Programmiersprache BASIC, elementare BASIC-Kommandos, Diskettenhandhabung und vieles, was zur perfekten Beherrschung des ATARI ST gehört.

Ein Spitzenbuch mit über 80 Übungs- und Anwenderprogrammen wie z. B.:

Sortierprogramme / Textverarbeitung / Umgang mit sequentiellen- u. Random-Dateien / Fakturierprogramm / Programmiertechniken an ausgewählten Beispielen u. v. a. m.

Zum Buch gibt es die Programmdiskette mit sämtlichen Beispiel-Programmen.



Buch: 49,- DM
Erscheint: Januar 1986

Das Buch für den richtigen Einstieg mit dem ATARI ST.

Leicht verständlich wird der Lernende in den Lernstoff eingeführt.

Einige der Themen:

die Hardware des ATARI ST
- Überblick über die Systemkomponenten
- Aufstellung des Computers
- Wartung

die Software des ATARI ST
- wie arbeite ich mit GEM (das Desktop / Maus / Icons etc.)
- die Programmiersprachen BASIC / LOGO
- die Programme GEMDRAW / GEMWRITE
- Kopieren von Files und Disketten, Löschen und Formatieren u. v. a. m.